Ransomware Mitigation in the Modern Era: A Comprehensive Review, Research Challenges, and Future Directions

TIMOTHY MCINTOSH, A. S. M. KAYES, YI-PING PHOEBE CHEN, and ALEX NG,
La Trobe University, Australia
PAUL WATTERS, Cyberstronomy Pty Ltd, Australia

Although ransomware has been around since the early days of personal computers, its sophistication and aggression have increased substantially over the years. Ransomware, as a type of malware to extort ransom payments from victims, has evolved to deliver payloads in different attack vectors and on multiple platforms, and creating repeated disruptions and financial loss to many victims. Many studies have performed ransomware analysis and/or presented detection, defense, or prevention techniques for ransomware. However, because the ransomware landscape has evolved aggressively, many of those studies have become less relevant or even outdated. Previous surveys on anti-ransomware studies have compared the methods and results of the studies they surveyed, but none of those surveys has attempted to critique on the internal or external validity of those studies. In this survey, we first examined the up-to-date concept of ransomware, and listed the inadequacies in current ransomware research. We then proposed a set of unified metrics to evaluate published studies on ransomware mitigation, and applied the metrics to 118 such studies to comprehensively compare and contrast their pros and cons, with the attempt to evaluate their relative strengths and weaknesses. Finally, we forecast the future trends of ransomware evolution, and propose future research directions.

CCS Concepts: • **Security and privacy** → **Operating systems security;**

Additional Key Words and Phrases: Ransomware, ransomware detection, ransomware defense, ransomware prevention

ACM Reference format:
Timothy McIntosh, A. S. M. Kayes, Yi-Ping Phoebe Chen, Alex Ng, and Paul Watters. 2021. Ransomware Mitigation in the Modern Era: A Comprehensive Review, Research Challenges, and Future Directions. *ACM Comput. Surv.* 54, 9, Article 197 (October 2021), 36 pages.
https://doi.org/10.1145/3479393

1 INTRODUCTION

In recent years, ransomware frequently dominated the headlines of cybercrime reports, when repeated episodes of ransomware attacks on both organizations and individuals have created massive financial losses and disruptions to normal lives (Table 1). Ransomware has evolved from simple

Authors' addresses: T. McIntosh, A. S. M. Kayes, Y. -P. Phoebe Chen, and A. Ng, La Trobe, La Trobe University, Plenty Rd & Kingsbury Dr, Bundoora VIC 3086, Australia; emalis: {t.mcintosh, a.kayes, phoebe.chen, alex.ng}@latrobe.edu.au; P. Watters, Cyberstronomy Pty Ltd, Ballarat, Victoria, Australia, 3350; email: ceo@cyberstronomy.com.

0360-0300/2021/10-ART197 $15.00
https://doi.org/10.1145/3479393

Table 1. A Brief History of Notable Ransomware Episodes

Year	Event
1989	The first known and documented ransomware: *AIDS Trojan* (also known as the *PC Cyborg* virus)
2006	*GPCode* and Archiveus Trojan began using more powerful asymmetric RSA encryption.
2013	*CryptoLocker*, one of the most damaging ransomware, possibly profited US$27 million
2014	*ScarePackage* ransomware was the first to target Android mobile platforms. It infected users as fake apps appearing to be Adobe Flash or other well-known antivirus, and pretended to scan user files when launched.
2015	*CryptoWall 2.0*, delivered via emails, PDF and various exploit kits, used TOR to obfuscate C&C communications, and incorporated anti-virtual-machine and anti-emulator techniques to avoid analysis via sandboxing.
2015	*Linux.Encoder* was considered the first known ransomware targeting the Linux platform.
2016	*KeRanger* was discovered as the first known ransomware targeting the MacOS platform.
2016	*PoshCoder* became the first fileless ransomware and instead used PowerShell commands to encrypt files.
2017	*WannaCry* attack propagated through the Windows EternalBlue exploits, causing an estimated USD$4billion of damage globally through loss of data and disrupted business processes.
2017	*NotPetya* was the first known ransomware to encrypt the Master File Table of the NTFS drive.
2020	*RagnarLocker* was the first known virtual-machine-based ransomware to be deployed as virtual machines and to encrypt host files via shared folders, in order to evade host-based ransomware detection.
2020	*Maze* was the first known ransomware to exfiltrate sensitive data to blackmail users into paying ransom.

scareware and ***User Interface*** (**UI**)-lockers, to cryptographic ransomware, and recently to fileless ransomware and ransomware with data exfiltration. Cryptocurrencies have further fueled the ransomware pandemics, and ransomware victims lacked control over the extortionists' decryption guarantees [89]. In 2021, ransomware attackers appeared to have shifted their targets from individuals to larger corporates, which is more likely to result in higher ransom profits; the damages caused by publicized ransomware attacks are expected to reach US$6 trillion, due to an increased number of more disruptive and sophisticated attacks.[1] Contrary to traditional malware, ransomware could cause irreversible damage to the ***Operating System*** (**OS**) or user files even after its removal [43, 106]. Therefore, thwarting ransomware attacks at the earliest possible opportunity becomes the major challenge in combating ransomware.

The increasing threat of ransomware has generated an exponential growth of research to analyze and mitigate ransomware attacks from different perspectives, such as *detecting* the presence of known ransomware or recognized traits of its attacks, *defending* the OSs and user files from unwanted modification or access, and *preventing* ransomware from reaching or attacking victims. Until the completion of this survey, many of the existing studies took either a programmatic or a data-centric approach, and were based on machine learning. Yet cybercriminals continue to find new means to circumvent current countermeasures; some have explored new attack vectors or new platforms that could easily defeat existing mitigation mechanisms. For example, fileless ransomware scripts would render any analysis of executable files useless. Making sense of the state of ransomware evolution against anti-ransomware research will enable us to review the anti-ransomware landscape, target the root causes of ransomware attacks, and propose better anti-ransomware mechanisms that can be more effective over a longer time.

Several surveys in the domain of ransomware have been published, all between 2019 and 2020 (in Supplementary Material, Section B). However, all of them appear to be either limited in scope (e.g., focusing on ransomware detection only, preferring programmatic solutions without considering user-centric ones, or favoring Microsoft Windows–based proposals), or making superficial comparisons (e.g., simply comparing the declared detection accuracy of those proposals without further insights). The objective of this survey is to provide a comprehensive review of the definition and classification of ransomware, the analysis of different variants and its future trends, and different detection, defense, and prevention strategies. While we aim to cover a broad spectrum of academic

[1] https://www.blackfog.com/the-state-of-ransomware-in-2021.

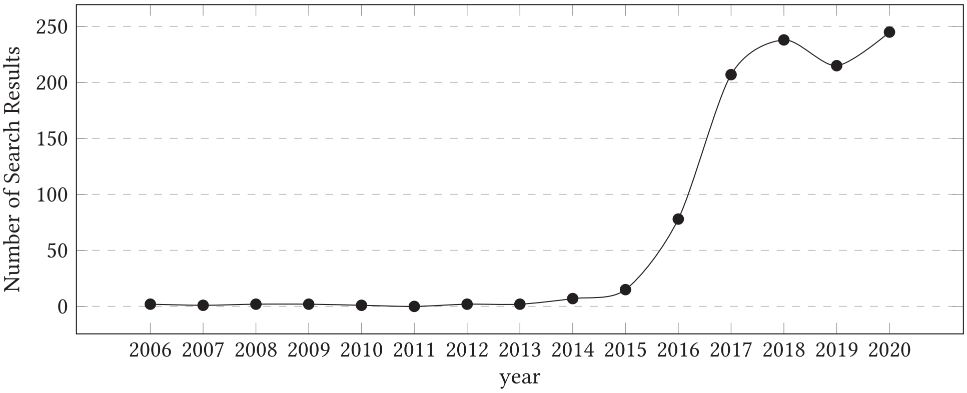


Fig. 1. The Number of articles with titles containing the keyword "Ransomware" per year on Google Scholar.

research, we focus on academic papers that are actual anti-ransomware proposals with completed theoretical foundations and evaluations with ransomware, but exclude incomplete proposals based only on assumptions or case studies, and studies that simply repeat one another with little impact.

The major contributions of this survey include the following:

- We reviewed and proposed updates to more accurately define ransomware-related terminologies: ransomware, ransomware detection, ransomware defense, ransomware prevention and ransomware mitigation.
- We reviewed 118 published anti-ransomware studies of very different methodologies and results, and proposed a set of unified metrics to attempt to validate the internal and external validity of those studies.
- We concluded a few common themes among published anti-ransomware studies, listed their limitations, and suggested future research directions to combat ransomware evolution into new attack vectors and attack patterns.

2 RESEARCH MOTIVATION

In this section, the background information to motivate this survey is presented. Ransomware is a relatively new attack vector but has generated significant research interest, when the number of articles (excluding patents and citations) containing the keyword "Ransomware" in their titles on Google Scholar have risen dramatically since the year of 2006 (Figure 1). Although no article was found on Google Scholar to contain "Ransomware" in its title prior to 2006, the ransomware-specific research seemed to have taken off since 2014. However, we have found a few issues that have not been properly covered in existing research.

2.1 Interchangeable Usage of Ransomware and Crypto-Ransomware

The first issue we noticed was the prevalent interchangeable usage of the terminologies *ransomware* and *crypto-ransomware*, which could suggest the absence of a consensus in the research community on whether those two terminologies are technically equivalent, or whether non-crypto-ransomware are considered ransomware at all. For example, a few studies (e.g., [24, 41, 46, 156]) specified "crypto-ransomware" or "cryptographic ransomware" as their research target, while others (e.g., [6, 42, 140]) simply referred to it as "ransomware" while still mentioning its cryptographic activities.

2.2 Lack of a Universal Standard to Define Benign or Malicious (Ransomware-Like) Behaviors

Ransomware is considered one class of malware, or malware with the primary aim to extort ransom payments from users [82]. The definition of *malware* is extensively attempted in literature: *malware* was defined in [88] as software that harmfully attacks other software with behaviors that are different from the intended behavior as understood and expected by the users. In [122], *malware* was considered as a program with malicious intent and may damage the machine or the network on which it operates. Similarly, there have been various different definitions of *ransomware* in literature. In [82], ransomware was one class of scareware to coerce victims into paying to re-gain access to their data. In [46], ransomware was defined as malware that removed authorized users' access to their data until ransom payments are made. By comparing the definition of malware with that of ransomware, we could safely assume that the consensus agreed in that different variants of ransomware tend to share two common features: (1) blocking user access to resources, often files, and (2) attempting to extort ransom payments. However, those studies declared different features or different combinations as malicious.

2.3 Lack of Uniform Usage of Terminologies on Mitigation Strategies

Many research articles chose to express the main intention of their research outcome by using one or more of the following terminologies: detection, defense, prevention, or mitigation. However, there seemed to be a lack of the uniform usage of the terminologies in ransomware analysis. In literature, malware *detection* is to identify the presence of concealed malware or its activities; malware *defense* is to defend from or to resist malware attacks when it is already in progress; malware *prevention* is to stop malware attacks from arising or taking place [80, 88]. However, such clear usage of terminologies appears to be less evident in ransomware mitigation.

2.4 No Universal Standards in Evaluating and Comparing the Effectiveness of Different Strategies

Many studies focused on ransomware detection claimed high detection rate and low error rate, despite wildly different design principles, testing regimes, and coverage of ransomware samples, yet their implementations and samples used are often not available for further validation or scrutiny by other researchers. Some studies suggested prevention or general precaution strategies, of which the efficiency often cannot be properly tested unless controlled trials can be performed.

3 RANSOMWARE ANALYSIS, CLASSIFICATION, AND TRENDS

In this section, we present our ransomware analysis, attempt to classify ransomware according to the chronological appearance of its variants, and predict future trends.

3.1 Ransomware Analysis

The primary intrusion principle of ransomware is to stealthily attack and take control of irreplaceable user resources (files and data), until it has gained full control of them, and has either excluded the user's access or jeopardized the user's exclusive access, before declaring its presence and demanding ransom payments [43, 106]. Initially, ransomware has to attack stealthily, to evade detection by antivirus and to allow sufficient time to complete time-consuming tasks (e.g., file encryption or information exfiltration) [43, 82]. Ransomware must target irreplaceable user resources, to increase the willingness of users paying ransom [33]. It must gain full control of user resources and jeopardize user's full access, to prevent users from getting out of paying ransom by simply repairing the OS or removing the ransomware itself [58, 81, 103]. Because this intrusion principle is common among ransomware, a key element in combating ransomware is to prevent ransomware from accessing those user resources, an issue that falls under the umbrella of proper access control. Since ransomware is a type of cyber intrusion, it is helpful to analyze its attacks using the ***Cyber-Kill-Chain*** (**CKC**) model. A CKC-based taxonomy has been extensively introduced in [46], which described the steps ransomware must take to perform attacks: (1) weaponization, in which cybercriminals plan how to launch ransomware attacks; (2) delivery, in which the ransomware payload must be transported to reach the victims; (3) exploitation, in which ransomware needs to be executed in the victim environment; (4) installation, in which ransomware gains and maintains the presence in the victim OS; (5) command and control, in which ransomware communicates with the cybercriminals for malicious tasks such as encryption key exchange and information exfiltration; and (6) actions on objectives, in which ransomware disrupts the victims and demands ransom payments. Although the survey of [46] was performed on studies mostly investigating cryptographic ransomware, its CKC model is also applicable to recent ransomware variants that do not solely perform file encryption. The CKC model guides researchers to collect comprehensive and structured information about ransomware attacks, to better understand ransomware behaviors, and to build more secure ransomware mitigation solutions.

3.2 Ransomware Classification by Evolution of Evasion Techniques

Many studies have attempted to classify ransomware in different ways, for example, by genealogy or binary similarity (e.g., [19, 45, 153, 155]), and by platform (commonly used by the industry). In this survey, we have decided to classify ransomware by evolution of evasion techniques employed by ransomware. Evasion by ransomware is a critical feature for the completion of ransomware attacks before full control of user resources is taken [81, 82, 130]. It was found that ransomware variants developed around the same time tended to employ similar evasion techniques. Therefore, this method of ransomware classification offers a progressive view on how ransomware evolves to evade detection solutions, bypass defense mechanisms, and circumvent prevention techniques.

In this survey, scareware is not considered one type of ransomware. *Scareware*, sometimes also known as hoaxware or fake ransomware, often presents a user interface to demand ransom payments by deception [30, 116]. It can either falsely inform users that they have been infected with a destructive malware, or masquerade as an antivirus product or technical support from security companies [30]. After users purchase the “removal” services, the scareware removes its presence. Scareware is usually purely nuisance and does not actually damage the infected system, and can be terminated by killing its process in the memory and removing startup entries, but it could still deceive some unsophisticated users and lead them to falsely believe they are protected by antivirus software. In [25], scareware was considered to be one type of ransomware, but this view has not been shared by most other researchers.

3.2.1 ScreenLocker. *ScreenLocker* can either be legitimate software that locks a computer system while the user is away, or malware that locks the user interface to blackmail a victim into paying ransom [82]. In the context of ransomware, *ScreenLocker* ransomware can lock the whole OS interface to disable user operations, force the user to stay on the current ransomware UI, or create repeated fake warnings to threaten users, often without actually attacking user files and data [80]. It is considered more damaging than scareware, as some variants implement mechanisms to maintain persistence in the system and to prevent it from being terminated easily. However, *ScreenLocker* ransomware can usually be easily removed by terminating the process and deleting the payload, with no loss to user files and data. *ScreenLocker* relied on the fear of unsophisticated users to demand ransom payments. Shortly after the appearance of *ScreenLocker*, there have appeared numerous tutorials teaching victims how to remove it.

3.2.2 Crypto-Ransomware. *Crypto-ransomware*, also known as *cryptographic ransomware*, is the variant of ransomware that employs strong cryptography to attack the file systems, prevents legitimate users from accessing important files and data, and demands ransom payments. The strong encryption it employs on user files or the MBR guarantees its exclusive access to them. It has recently become the predominant type of ransomware, accounting for 90% of the attacks in 2019.[2] Possibly due to the dominance of *crypto-ransomware* in the ransomware landscape, some studies use it interchangeably with ransomware, or consider it the only type of ransomware (Subsection 2.1). While most crypto-ransomware variants attack selected file types [106], a few variants such as *Petya* ransomware attacks the ***Master File Table*** (**MFT**) of the file system [106]. Crypto-ransomware must attack the file system to be able to hold user files and data hostage [80, 106]. Crypto-ransomware was found to perform more activities of cryptography, file system and network communications, and often more frequently than benign applications [59]. Users are generally advised against making ransom payments to cybercriminals, because (1) any payments will fund further criminal activities, (2) not all paying victims get the decryption keys, and (3) not all decryption keys work to get victim files and data decrypted [33, 42].

Because of the rapid increase in crypto-ransomware campaigns, the number of academic studies and industry solutions against crypto-ransomware has ballooned. The latest Windows 10 is equipped with Ransomware Guard,[3] whereas many other major antivirus vendors have included anti-ransomware modules in their products (e.g., ESET Ransomware Shield[4]) to specifically detect file encryption behaviors, although the implementation details of commercial antivirus remain their trade secrets. However, over time, it has become more challenging for a ransomware executable file to perform file encryption on user files, forcing ransomware developers to seek better evasion techniques.

3.2.3 Fileless Ransomware. *Fileless ransomware* is the form of ransomware that executes without first placing the traditional form of malicious executables on the file systems of the victims, but delivering the payload via alternative means such as *Command Scripts* (e.g., JavaScript, PowerShell, batch commands) or via ***Remote Desktop Protocol*** (**RDP**) connections. Fileless ransomware was developed to evade the ever increasing scrutiny aforementioned on executable files performing user file and data encryption. In [100], the evolution of fileless malware attacking the computer systems without a malicious executable file (e.g., the *exe* files on Windows platforms) was discussed, but only the malware attacks by malicious command scripts were described. First seen in 2014, *Poshcoder* was the first fileless ransomware that mimicked Locky ransomware by using PowerShell scripts on Windows OS.[5] Ransomware like *SamSam* and *CryptON* attacked via RDP connections, and RDP-based targeted ransomware attacks are becoming more common [148]. Because there is no malicious executable on the local file system, the traditional signature-based malware scans cannot detect its presence. Behavioral-based malware detection may flag its container (e.g., the PowerShell execution environment) as benign and fails to detect the malicious code itself.

3.2.4 VM-Based Ransomware. *VM-Based Ransomware* is when the ransomware deploys a VM on the host system and hides in it to avoid host-based malware detection, maps host folders as shared folders into the VM, and performs encryption of host files in the VM. It was developed to evade detection on any non-OS code performing file encryption, whether there was a known

[2] https://knowledge.broadcom.com/external/article?legacyId=HOWTO124710.

[3] https://support.microsoft.com/en-us/windows/protect-your-pc-from-ransomware-08ed68a7-939f-726c-7e84-a72ba92c01c3.

[4] https://help.eset.com/glossary/en-US/technology_ransomware_protection.html.

[5] https://blog.trendmicro.com/trendlabs-security-intelligence/ransomware-now-uses-windows-powershell.

executable file involved or not. The *Ragnar* ransomware was the first of such kind, by hiding inside a Windows XP VM with minimal modules. At the moment, no academic research has been found to propose countermeasures against *VM-Based Ransomware*. The emergence of *VM-Based Ransomware* has raised the game of malware evasion by applying sandboxing in an offensive manner, but it requires the hardware support of virtualization and the manual deployment of a virtual machine, a task that is difficult to automate. Ransomware researchers would need to think outside the box and consider countermeasures beyond pure detection on suspicious executables.

3.2.5 Ransomware with Data Exfiltration. *Ransomware with Data Exfiltration* is when ransomware no longer blackmails users with loss of access to data, but instead with a potential data leak of sensitive data. It was developed to evade the ever increasing scrutiny on any file encryption behavior, and the more robust backup schemes. The *Maze* ransomware and the *DoppelPaymer* ransomware are two such examples. At the moment, no academic research has been found to propose countermeasures against ransomware with data exfiltration. There are now even reports of "triple extortion" by ransomware, when cybercriminals either launch additional attacks (e.g., ***Distributed Denial of Service*** (**DDoS**)),[6] or blackmail data owners (e.g., patients who have confidential medical records stored with the hospitals) with privacy breach.[7] The emergence of *Ransomware with Data Exfiltration* shows that file encryption is only an optional attack vector and not the final goal, but the process of blackmailing victims for ransom payment is.

3.3 Ransomware Future Trends

As the combat against ransomware intensifies, ransomware developers are constantly looking for ways to evade existing ransomware mitigation techniques. Earlier versions of ransomware usually abused the vulnerabilities that allowed them to perform their critical task (i.e., taking control of critical user resources, and excluding or jeopardizing user's exclusive control) in the easiest manner. As the academic and industry communities move to mitigate such vulnerabilities used by ransomware, ransomware developers would naturally seek the next easiest evasion techniques. However, what remains the same is the critical task of ransomware aforementioned. Therefore, the armed race against ransomware becomes an issue of access control: ransomware aims to bypass existing access control of critical user files and resources, whereas anti-ransomware proposals seek to maintain access control of critical user files and resources, to prevent any exclusion or jeopardy by ransomware.

In the recent high-profile ransomware attacks, the attackers seem to have even further revolutionized their attack methods. Some previous studies attempted to predict how ransomware could "improve" technically, such as applying more complicated encryption schemes (e.g., [54, 121]), and attacking other platforms (e.g., [31]). However, such improvements may only be realized if ransomware developers decide to still perform file encryption, which has proven to be no longer necessary to extort ransom payments from victims. The key focus of ransomware evolution is to evade mitigation, so ransomware developers would have to explore less known attack vectors. The following trends have been observed in recent news reports and have not been included in any academic research:

- Less attacks on private individuals and more on enterprises, demanding larger sums of ransom payments.
- Less passive infiltrations (e.g., via phishing) and more active intrusions (e.g., hackers seeking systems security vulnerabilities).

[6] https://www.securitymagazine.com/articles/95238-welcome-to-the-new-world-of-triple-extortion-ransomware.
[7] https://www.techrepublic.com/article/ransomware-attackers-are-now-using-triple-extortion-tactics.

- —Shifting the focus from file encryption to damaging or disrupting victims in other ways (e.g., sensitive information exfiltration, DDoS attacks).

Therefore, future anti-ransomware research may need to consider those trends, focus on victim protection, and aim to minimize damage or disruption by ransomware on the victims.

4 UNIFIED METRICS IN EVALUATING STUDIES ON RANSOMWARE MITIGATION

In this section, we aim to explore the open issues of research in ransomware mitigation, and the possibility of establishing a set of unified metrics in evaluating those published works. Researchers have attempted to approach the issue of ransomware mitigation, and have produced an abundance of research papers. However, the landscape of ransomware has been constantly evolving, and so is the armed race between cybercriminals and ransomware researchers. As the knowledge of ransomware by researchers grows, ransomware is better understood and the research on ransomware attack mechanisms and mitigation strategies becomes more relevant. For example, the percentage of screen-locker ransomware is gradually declining, giving way to cryptographic ransomware [41, 46, 121]. Mitigation strategies focusing on detecting screen-locking behaviors, such as [18, 138], may become less effective over time against ransomware that does not lock user UI. Previously it was thought that ransomware must contact its C&C to generate unique encryption keys for each victim [2, 144]. However, recently it was found that 33.9% of ransomware variants did not make contact to C&C during attack phase [27], making such network-based ransomware detection on traffic to C&C less effective. Although earlier studies have contributed to the knowledge of anti-ransomware research, some of them are no longer adequate or accurate in combating the most recent ransomware threats.

We believe the evaluation result of a study depends on the combination of all these factors: the algorithm applied, malicious features extracted, software programming implementation and settings of their proposal, quality and quantity of ransomware samples used, the rigorousness of the evaluation process, and the training process (if machine learning is applied). While it is up to the readers to interpret the evaluation results of anti-ransomware research, there clearly lacks a set of universally agreed on standards in evaluating the effectiveness of different research outcomes. Therefore, it is unfair to simply compare the accuracy of one study against those of other studies, without considering other confounding factors. After evaluating the current challenges in assessing the existing ransomware mitigation studies, we have proposed a set of unified metrics in evaluating and comparing existing studies on ransomware mitigation (Table 2), to help us determine which studies have survived better over time and are more resilient against ransomware evolution. The metrics were grouped into four groups:

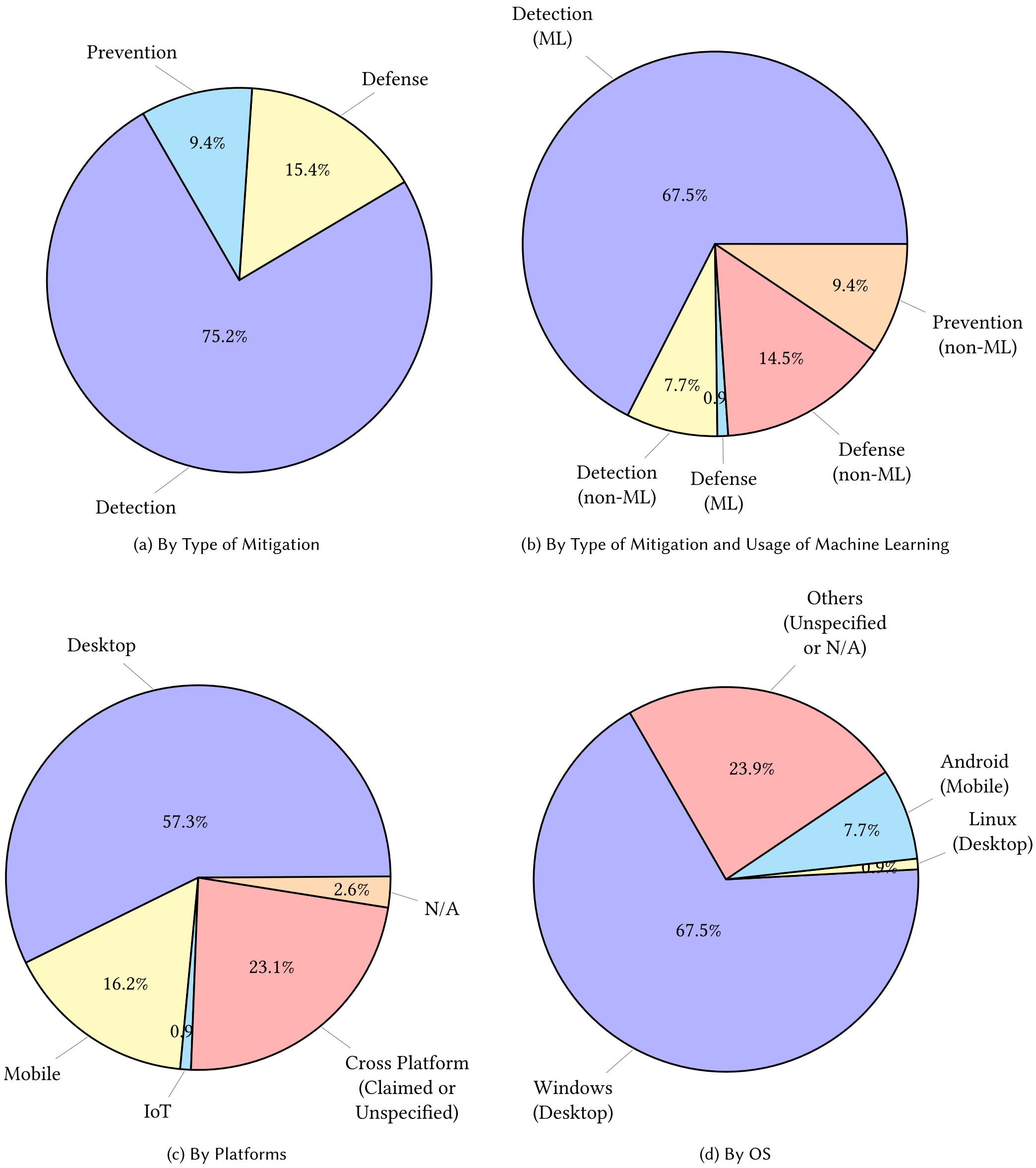
- —Evaluation: to assess how well the existing anti-ransomware proposals were properly evaluated themselves. More robust studies are more likely to rigorously evaluate the internal and external validity of their studies.
- —Output: to assess how informative the studies are and how much they have contributed to the knowledge of anti-ransomware research.
- —Versatility: to assess how versatile the studies are in facing the emerging new trends and challenges of the ransomware landscape.
- —Strengths: to assess the additional strengths of anti-ransomware solutions.

5 RANSOMWARE MITIGATION STRATEGIES

In this section, we compare and contrast the different ransomware mitigation strategies. There are three main approaches to mitigate ransomware attacks (Figure 2): *detection*, *defense* and *prevention* (See definitions in Supplementary Material A). Ransomware *detection* is proactive, when the implementations can identify the ransomware or its attack activities before it completes any

Table 2. Unified Metrics for Evaluating Studies on Ransomware Mitigation

Category	Criteria	Description and Rationale
Evaluation	Evaluated with Known Ransomware	It should be an important criterion to test the effectiveness of such a study [80, 81].
	Evaluated with Unknown Ransomware	The fight against ransomware is an armed race, and anti-ransomware implementations should aim to get ahead of the game [43, 81, 103]. Unknown ransomware includes simulated activities, and samples used for the testing phase of machine learning.
	Evaluated with Benign Applications	As anti-ransomware implementations should aim to achieve high detection rate while minimizing the false-positive rates against benign applications, a good implementation should be evaluated with benign applications, including consented user-initiated operations, to demonstrate its practicality via low false-positive rates [43].
	Evaluated against Competitors	A rigorous anti-ransomware implementation should attempt to compare itself with competitors, which may include existing studies and industry antivirus software products, and aim to prove its value by demonstrating its superiority over competitors.
Output	Malicious Features Specified	An informative study should consider listing the malicious features of ransomware it is trying to mitigate.
	Causal Relationships Analyzed	A meticulous anti-ransomware study should discuss the causal relationship between malicious features observed and the actual malicious behaviors of ransomware.
	Accuracy Discussed	A mature and punctilious anti-ransomware study should discuss its accuracy, either as the accuracy of ransomware detection, or as the percentage of ransomware attacks it is likely to defend against or to prevent.
	Efficiency Discussed	A carefully designed anti-ransomware study should discuss its efficiency of performing ransomware mitigation, and should aim to achieve the highest possible efficiency to enable practical use, as some variants of ransomware could aggressively encrypt a large quantity of user files within a very short duration before it is terminated [106]. On mobile platforms, efficiency may also need to include battery consumption due to intensive computing.
Versatility	User Files Protected	Whether the anti-ransomware proposal could protect user files from ransomware attacks.
	Boot Sector Protected	Whether the proposal could protect the boot sector of the storage device from ransomware attacks.
	Data Exfiltration Prevented	Whether the proposal could prevent exfiltration of private or sensitive data by ransomware
	Effective against Other Attack Vectors	Whether the proposal could be effective against other attack vectors seen recently, such as fileless ransomware scripts, RDP -based or VM-based ransomware.
	Cross Platform Supported	Whether the proposal is capable of mitigating ransomware on multiple platforms (*i.e.* desktop, mobile and IoT).
	Architecture Independent	Whether the proposal is independent of specific architectures used in computer systems. For examples, anti-ransomware implementations detecting HTTPS-based web traffic are specific to the architecture of the HTTPS protocol, and may not work on ransomware generating web traffic using other protocols or those not generating web traffic.
Strengths	Self Disclosure of Limitations	The self disclosure of limitations of the research itself and suggestions of future research directions should be included in the published manuscript, as it is unlikely that any study could be perfect [123].
	Minimal Delay until Full Effect	Whether the proposal can reach full mitigation effect with minimal delay of time spent collecting information or performing analysis of ransomware. If an anti-ransomware study could achieve its full effect with minimal delay, it could potentially minimize the quantity of user files and resources attacked or exfiltrated by ransomware [106].
	Difficult to Evade	How difficult for ransomware to evade the mitigation by the anti-ransomware proposal.



(a) By Type of Mitigation

(b) By Type of Mitigation and Usage of Machine Learning

(c) By Platforms

(d) By OS

Fig. 2. Statistics of existing anti-ransomware proposals.

irreversible damages. Ransomware *defense* is reactive, when the attempts by ransomware to perform irreversible damages are thwarted or reversed. Ransomware *prevention* is preemptive, in which the infection or the execution of ransomware is prevented from taking place. Table 3 further shows in detail where different implementations within each strategy fit in with the overall OS architecture, classified based on their primary research methods.

5.1 Ransomware Detection

This subsection lists and compares the studies to detect the presence of ransomware or the progress of ransomware attacks.

Table 3. Where Different Ransomware Mitigation Implementations Work in OS Architecture

		Detection		Defense	Prevention
Users, System Adminsm and Organizations	User Experience		Displayed Ransom Message		User Education and Awareness [26, 97, 143]
	Policies, Procedures, Guidelines		Event Log Monitoring, Auditing, PenTest		Company Policies on SecureOps [69, 70]
User Files and Other Data	File and Data Maintenance		Loss of Access to User Files and Data	Cloud Storage [34, 61, 90], File Backup [71]	File Access Control [17, 93], Ransomware Protected Folder (Trend Micro, BitDefender, Microsoft Defender)
	Physical Files and Data, Settings		Honeypot Files [50, 58, 113]	OS Hardening [150]	Application Whitelisting [84, 145]
User Mode	User Applications	ML	Static Analysis [10, 13, 15, 16, 39, 45, 48, 74, 79, 98, 107, 127, 138, 154, 155], App Behavioral Analysis [1, 3, 14, 18, 35–37, 64, 65, 95, 157], Static and App Behavioral Analysis [52, 57, 60, 75, 134]	AntiVirus Real-Time Monitoring	
		non-ML	AntiVirus Scans, Static Analysis [73], App Behavioral Analysis [137], Static and App Behavioral Analysis [151]		
	Subsystems (e.g., Win32, POSIX, Android Runtime)	ML	API or Opcode Runtime Analysis [4, 7, 9, 21, 40, 94, 99, 110, 131, 133, 141], Detection of OS Compromise [78, 132, 146, 158], Encryption Detection [8, 62, 85, 86], File System Activity Analysis [20, 63, 66, 91, 130], Network Activity Analysis [12, 28, 32, 44, 51, 112, 124, 129, 144], Sandboxing [142]	Encryption Key Interception [23, 47, 55, 56, 87, 92, 119], OS Resilience [105]	
		non-ML	Detection of OS Compromise [125], File System Activity Analysis [38, 72, 83, 101, 102, 118], Network Activity Analysis [2, 114, 148, 149], Sandboxing [80]		
Kernel Mode	Executive (e.g., I/O Manager, Process Manager, Memory Manager)		Monitoring OS Kernel Activities [43, 81, 108]		
	Kernel Mode Drivers		Windows Malicious Software Removal Tool		Windows Trusted Installer
	Hardware Abstraction Layer				Virtualization
Hardware	Firmware		Storage Buffer Management [117]	SSD Firmware-Based Recovery [22, 68, 111, 147]	Firmware-Based Access Control [5, 136]
	Physical Device		HPC Activities Analysis [11]	Backup Using Stealthy Space [139]	

5.1.1 *Hardware Level.*

At the hardware level (Table 4), ransomware detection can be implemented either on the hardware sensors or on the firmware level of the hardware devices. Ratafia was presented in [11] as a two-step unsupervised machine learning solution on the HPC events (branch instructions, branch misses, cache references, and cache misses), using both Deep Neural Network and Fast Fourier Transformation. Despite the novel method of analyzing HPC values for possible ransomware activities, it is a low-level snapshot of microprocessor activities, making it unable to pinpoint HPC events to one particular process being executed. In [117], ransomware-aware buffer management of the internal hardware of storage devices was proposed, to identify access patterns to storage devices that were similar to those of crypto-ransomware investigated, but the proposal was evaluated using synthetic disk request patterns and was only tested with *CryptoHasYou* ransomware. Although the hardware-level implementations can capture low-level features and access patterns, they usually do not have access to semantic information (i.e., filenames, OS activities, application behaviors) and could only speculate based on the hardware access

Table 4. Comparison of Different Studies on Ransomware Detection

Category	Research	Project Name	Year	Relevance			Evaluation				Output				Versatility						Strengths		
				Machine Learning Based	Peer Reviewed	Cited by Others	with Known Ransomware	with Unknown Ransomware	with Benign Appliction	against Competitors	Malicious Features Specified	Causal Relationships Analyzed	Accuracy Discussed	Efficiency Discussed	User Files Protected	Boot Sector Protected	Data Exfiltration Prevented	Effective Against Other Vectors	Cross Platform	Architecture Independent	Self Disclosure of Limitations	Minimal Delay until Full Effect	Difficult to Evade
Hardware Level	[11]	Ratafia	2019	Y	Y	5	✓	×	✓	✓	✓	✓	×	✓	△	×	×	×	✓	×	✓	×	×
	[117]		2019	N	Y	0	△	×	×	×	✓	✓	×	×	△	×	×	×	×	×	×	×	×
Kernel Mode Level	[43]	ShieldFS	2016	Y	Y	149	✓	×	✓	×	✓	✓	✓	✓	✓	×	×	△	×	×	✓	✓	△
	[81]	Redemption	2017	Y	Y	72	✓	×	✓	×	✓	✓	✓	✓	✓	△	×	×	△	×	✓	✓	△
	[108]	RWGuard	2018	Y	Y	29	✓	×	✓	✓	✓	✓	✓	✓	✓	×	×	△	△	×	△	△	×
API or Opcode Runtime Analysis	[110]		2016	Y	Y	87	✓	×	✓	✓	×	×	✓	×	△	×	×	×	×	×	×	×	×
	[99]		2017	Y	Y	16	✓	×	✓	×	×	×	△	×	△	×	×	×	△	×	×	×	×
	[40]		2018	Y	Y	38	✓	×	✓	×	✓	✓	✓	×	×	×	×	×	×	×	✓	×	×
	[141]		2018	Y	Y	23	✓	×	✓	×	×	×	✓	×	△	×	×	×	×	×	×	×	×
	[9]		2018	Y	Y	8	✓	×	✓	×	×	×	✓	×	△	×	×	×	×	×	×	×	×
	[21]		2018	Y	Y	4	✓	×	✓	✓	×	×	✓	✓	△	×	×	×	×	×	×	×	×
	[131]		2019	Y	Y	11	✓	×	✓	✓	✓	×	✓	✓	✓	×	×	×	×	×	✓	×	×
	[133]		2020	Y	Y	3	✓	×	✓	✓	✓	✓	✓	×	△	×	×	×	×	×	×	×	×
	[94]		2020	Y	Y	0	✓	×	✓	×	✓	×	×	×	×	×	×	×	×	×	×	×	×
	[4]		2020	Y	Y	1	✓	×	✓	✓	×	×	✓	✓	△	×	×	×	×	×	×	△	×
Detection of OS Compromise	[132]	EldeRan	2016	Y	N	122	✓	✓	×	✓	✓	×	✓	×	✓	×	×	×	×	×	✓	×	×
	[146]		2018	Y	Y	3	✓	×	×	△	✓	×	✓	×	△	×	×	×	×	×	×	×	×
	[158]		2020	Y	Y	0	✓	×	×	✓	✓	×	✓	△	△	×	×	×	×	×	✓	×	×
	[125]		2020	N	Y	0	✓	✓	✓	✓	✓	✓	✓	×	✓	×	×	×	×	×	✓	△	×
	[78]		2020	Y	Y	0	×	△	✓	×	✓	×	✓	×	△	×	×	×	×	×	×	×	×
Encryption Activity Detection	[62]	CryptoKnight	2018	Y	Y	3	△	×	×	×	✓	✓	×	×	×	×	×	×	×	×	△	×	×
	[85]		2019	Y	Y	10	✓	×	✓	△	△	×	✓	×	△	×	×	×	×	×	×	×	×
	[8]		2020	Y	Y	6	✓	×	✓	✓	✓	✓	✓	×	✓	×	×	×	×	×	×	×	×
	[86]		2020	Y	Y	0	✓	×	✓	×	×	×	✓	×	✓	×	×	×	×	×	×	×	×
	[7]		2020	Y	Y	1	✓	×	✓	✓	×	×	✓	×	△	×	×	×	×	×	×	×	×
File System Activity Analysis	[130]	CryptoLock	2016	Y	Y	280	✓	✓	✓	×	✓	✓	✓	✓	✓	×	×	△	×	△	✓	✓	×
	[102]		2016	N	Y	34	△	×	△	×	✓	✓	✓	×	✓	×	×	×	✓	×	✓	✓	×
	[118]	Malware-O-Matic	2017	N	Y	14	✓	×	✓	×	✓	✓	✓	✓	✓	×	×	✓	✓	×	✓	×	×
	[83]		2018	N	Y	7	△	×	△	×	✓	✓	×	×	✓	×	×	✓	×	×	×	×	×
	[66]		2018	Y	Y	4	△	×	△	×	✓	△	✓	×	△	×	×	×	×	×	✓	✓	×
	[72]		2018	N	Y	14	△	×	×	×	✓	✓	×	×	△	×	×	△	×	×	×	×	×
	[38]		2019	N	Y	2	△	×	×	×	✓	✓	×	△	×	×	×	△	×	×	×	△	×
	[91]		2019	Y	Y	11	△	×	△	✓	✓	✓	✓	×	✓	×	×	✓	×	✓	×	✓	×
	[101]		2019	N	Y	1	△	×	△	×	✓	✓	×	×	✓	×	×	×	×	×	×	✓	×
	[63]		2019	Y	Y	0	△	×	△	×	✓	×	✓	×	△	×	×	×	△	×	×	×	×
	[20]		2020	Y	Y	0	✓	×	✓	×	×	×	✓	×	△	×	×	△	△	×	✓	×	×
Network Activity Analysis	[2]		2015	N	Y	65	✓	×	×	×	✓	✓	✓	×	×	×	△	△	✓	✓	✓	×	×
	[144]		2016	Y	Y	18	✓	×	×	×	✓	×	✓	×	×	×	△	△	✓	×	×	×	×
	[28]	DECANTeR	2017	Y	Y	17	✓	×	✓	✓	✓	✓	✓	×	×	×	✓	✓	✓	×	✓	×	×
	[32]		2018	Y	Y	80	△	×	△	×	✓	✓	✓	×	×	×	△	△	✓	✓	×	×	×
	[124]	Raptor	2018	Y	N	8	△	×	×	×	✓	×	✓	×	×	×	△	△	✓	✓	✓	×	×
	[44]		2018	Y	Y	24	✓	×	✓	×	×	×	✓	×	×	×	△	✓	✓	✓	×	×	×
	[148]		2018	N	Y	19	✓	×	×	×	✓	✓	×	×	×	×	△	△	×	×	×	×	×
	[114]	Redfish	2018	N	Y	24	✓	×	×	✓	✓	✓	✓	✓	✓	×	✓	△	✓	✓	✓	△	×
	[51]		2019	Y	Y	11	✓	×	✓	✓	✓	×	✓	△	×	×	✓	✓	✓	✓	×	△	×
	[12]		2019	Y	Y	17	△	×	✓	×	✓	✓	✓	×	×	×	△	△	✓	✓	×	×	×
	[149]	IoTSDN-RAN	2020	N	Y	1	✓	×	×	△	✓	✓	✓	×	×	×	△	△	✓	✓	×	×	×
	[112]		2020	Y	Y	0	△	×	△	×	×	×	✓	×	×	×	△	×	✓	✓	×	×	×
	[129]	DeepRan	2020	Y	Y	0	✓	×	×	×	×	×	✓	×	×	×	×	×	✓	×	×	×	×

(Y: Yes; N: No; ✓: available; ×: unavailable; △: partially available).

patterns, which could come from multiple processes during the same time window. Ransomware could also obscure encryption activities to reduce the likelihood of CPU activities becoming out of the ordinary [103, 120]. Therefore, pure hardware-based ransomware detection may be inadequate in dealing with the complicated ransomware landscape.

5.1.2 Kernel Mode Level. On the level of OS *Kernel Mode* (Table 4), many ransomware detection implementations chose to augment with the executive layer of the OS to capture suspicious ransomware-like activities and OS information for further analysis. The implementations at the kernel model level ([43, 81, 108]) all leveraged OS-specific features to intercept ransomware-like file system activities for analysis. However, they could neither verify whether any activities were user-initiated, nor could intercept activities bypass the file system stack (i.e., direct disk access to physical sectors or modifications to MBR). All of them aimed for a one-shot detection at the earliest opportunity, and cannot fully rule out evasion of their detection. From a technical point of view, implementing solutions at the kernel level requires a comprehensive understanding of the OS architecture and demands advanced programming skills of the specific OSs, and it could be challenging to debug kernel programs, possibly making it less frequently used in research.

5.1.3 User Mode Level. Most of the technical implementation of ransomware detection appears to be operating on the level of *User Mode*, possibly because ransomware itself is more likely to run in user mode without attacking OS kernels [106], and because such implementations do not usually require complicated OS kernel programming.

API or Opcode Runtime Analysis. (Table 4): *Application Programming Interfaces* (APIs) are the standardized or publicized software interfaces that specify how they can be used by external modules to perform their functionalities, and in what formats data should be passed on as parameters [119, 131]. In many modern OSs, file system manipulations, a prerequisite for ransomware attacks, can only be performed via ransomware calling file system APIs offered by the OS [106, 119]. Opcode, short for operation code, is the machine instruction code translated from the executable files. The execution of ransomware and benign applications would inevitably create API calls, which will eventually be translated into opcode [153, 155]. Therefore, the usage patterns of the file system APIs of the OS may reveal ransomware-like activities on the file system. Several studies ([4, 9, 21, 40, 94, 99, 110, 131, 133, 141]) performed *API or Opcode Analysis* to detect ransomware. [131] predetermined certain API calls or opcode more likely to be malicious based on the nature of them , whereas others ([4, 9, 21, 40, 94, 99, 110, 133, 141]) concluded some API calls or Opcodes to be malicious after their machine learning algorithms found higher correlations between them and ransomware samples . The API or opcode-based detection methods can suffer from one or more of the following issues: (1) not explaining what API calls were considered malicious, or justifying why they were relevant to ransomware attacks; (2) not considering the parameters of such API calls or opcodes; (3) not specifying the boundaries of measuring API calls, while call stacks can be very tall; and (4) not providing solutions for circumvention of certain API calls, when ransomware could implement its own functions, or implement native C/C++ functions to bypass Java-based detection.

Detection of OS Compromise. (Table 4): Ransomware, as a type of malware, has to be executed in the OS to perform attacks, and often has to compromise the OS, depending on how its malicious payload is to be delivered. Signs of such compromise can include visual display of ransom messages, unknown automatic startup programs, unexpected access to Windows Registry, and unusually persistently high CPU usage [78, 132, 146]. Some studies ([78, 132, 146, 158]) explored how to analyze OS indicators for signs of compromise by ransomware using machine learning, whereas one study ([125]) did so without machine learning. While monitoring for possible indicators of

OS compromise could provide some clue of system anomaly, it needs to address the following issues: (1) how to properly define all finite states of the OS to accurately describe what is normal or abnormal; (2) how to accurately extract such indicators and to establish their relevance causation of ransomware attacks; (3) how to extract new indicators from zero-day attack vectors; and (4) how to determine whether such "OS compromise" was premeditated by users.

Encryption Activity Detection. (Table 4): For ransomware performing file encryption activities, the detection of such activities, either as cryptographic primitives in its payload, or the runtime API calls, may indicate its presence. Some studies ([7, 8, 62, 85, 86]) detected encryption activities in the OS to predict ransomware attacks. [62, 85] looked for cryptographic primitives in applications, whereas [7, 8, 86] analyzed application runtime for possible encryption activities. Detection encryption activities only works for cryptographic ransomware, and does not address the following issues: (1) not all ransomware variants encrypt user files directly; (2) encryption can be obscured, i.e., not using OS cryptographic libraries, performing using benign containers (e.g., PowerShell, VM, compression utility software); (3) not all ransomware variants perform encryption; and (4) benign software can also perform encryption and other cryptographic activities, so it is essential to differentiate defensive encryption and offensive encryption.

File System Activity Analysis. (Table 4): Ransomware attacking the file system usually causes changes in the target files, so the presence of file system activities by ransomware (e.g., massive number of file creation, deletion, file content, or type changes) would indicate ransomware attacks. Some studies ([20, 63, 66, 91, 130]) analyzed file system activities with machine learning to detect ransomware, whereas other studies ([38, 72, 83, 101, 102, 118]) analyzed file system activities without machine learning. Such studies made advancements in detecting ransomware that aggressively encrypt user files, but they failed on one or more of the following issues: (1) not all ransomware encrypts user files directly, i.e., some attacked MBR, some encrypted user files via command containers or virtual machine, and some performed sensitive data exfiltration instead; (2) analyses on benign or malicious file system activities were often based on known benign and ransomware samples, without critically analyzing the causal relationship between certain types of file system activities and ransomware attacks; (3) without user involvement, they were unable to differentiate user-initiated file encryption or destruction from ransomware-initiated ones; and (4) network-based or cloud-based storage may not provide mechanisms to assist in analysis on file system activities.

Network Activity Analysis. (Table 4): Ransomware, as a type of malware, may initiate outbound network traffic, often to random obscured domains or IP addresses, to communicate with remote cybercriminals, to generate unique encryption keys, or to exfiltrate sensitive information. A few studies ([12, 28, 32, 44, 51, 112, 124, 129, 144]) implemented network analysis by machine learning to detect ransomware-initiated network traffic. Other studies ([2, 114, 148, 149]) detected ransomware-initiated network traffic without using machine learning. While such studies contributed to the knowledge of ransomware research, they failed to address the following issues: (1) some ransomware variants did not initiate network traffic; (2) some ransomware-initiated network traffic could be stealthy, e.g., using obscured network protocols or via sandboxed virtual machines; and (3) it is not always possible to properly intercept and analyze all network traffic, especially on some mobile OSs.

Sandboxing. (Table 5): Sandboxing refers to the technique to execute malware in a controlled artificial environment that mimics a real end-user OS but is isolated from the OS, to observe and analyze the activities of the malware [80]. Two studies ([80, 142]) detected ransomware in

Table 5. Comparison of Different Studies on Ransomware Detection (Continued)

Category	Research	Project Name	Year	Relevance			Evaluation				Output				Versatility						Strengths		
				Machine Learning Based	Peer Reviewed	Cited by Others	with Known Ransomware	with Unknown Ransomware	with Benign Application	against Competitors	Malicious Features Specified	Causal Relationships Analyzed	Accuracy Discussed	Efficiency Discussed	User Files Protected	Boot Sector Protected	Data Exfiltration Prevented	Effective Against Other Vectors	Cross Platform	Architecture Independent	Self Disclosure of Limitations	Minimal Delay until Full Effect	Difficult to Evade
Sandboxing	[80]	Unveil	2016	N	✓	225	✓	✓	✓	×	✓	✓	✓	×	✓	×	×	×	×	×	×	✓	✓
	[142]	RansomSpector	2020	N	✓	0	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	×	×	×	×	×	✓	✓
Static Analysis	[73]		2017	N	Y	8	×	×	✓	×	✓	✓	×	×	△	×	×	×	×	×	×	×	×
	[98]	R-PackDroid	2017	Y	Y	52	✓	✓	✓	×	×	×	✓	×	✓	△	×	×	×	✓	✓	×	×
	[74]		2017	Y	Y	10	✓	✓	✓	×	×	×	✓	×	✓	△	×	×	×	×	×	×	×
	[16]	RanDroid	2018	Y	Y	6	✓	×	✓	×	×	×	✓	×	✓	△	×	×	×	×	×	×	×
	[45]		2018	Y	Y	1	✓	×	✓	×	×	×	✓	×	△	△	×	×	×	×	×	×	×
	[107]		2018	Y	Y	1	✓	×	✓	✓	✓	✓	✓	×	△	×	×	×	×	×	×	×	×
	[39]	Talos	2018	Y	Y	38	✓	×	✓	✓	✓	✓	✓	×	×	×	×	×	×	×	✓	×	×
	[138]		2018	Y	Y	13	✓	×	✓	✓	✓	✓	✓	×	×	△	×	×	×	×	✓	✓	×
	[15]	RanDetector	2019	Y	Y	0	✓	×	✓	✓	×	×	✓	×	✓	△	×	×	×	×	×	✓	×
	[155]		2019	Y	Y	52	✓	×	×	×	×	×	✓	×	△	△	×	×	×	×	×	×	×
	[10]		2019	Y	Y	11	✓	×	✓	×	×	×	✓	×	△	△	×	×	×	×	×	×	×
	[13]		2019	Y	Y	5	✓	×	✓	✓	✓	×	✓	×	✓	×	✓	×	×	×	×	×	×
	[154]		2019	Y	Y	1	✓	×	✓	×	×	×	✓	×	△	×	×	×	×	×	✓	×	×
	[48]		2020	Y	Y	1	✓	×	✓	×	✓	×	✓	×	✓	×	×	×	×	×	×	×	×
	[79]	DNAact-Ran	2020	Y	Y	1	✓	×	✓	×	×	×	✓	×	△	×	×	×	×	×	×	×	×
	[127]		2020	Y	Y	0	✓	×	✓	×	×	×	✓	×	△	×	×	×	×	×	×	×	×
App Behavioral Analysis	[18]	HelDroid	2015	Y	Y	151	✓	×	✓	×	×	×	✓	✓	✓	×	×	×	×	×	✓	△	×
	[137]		2016	N	Y	80	×	✓	×	✓	✓	✓	×	△	✓	×	×	×	×	×	×	△	×
	[1]	2entFox	2016	Y	Y	45	✓	✓	✓	✓	✓	✓	✓	×	△	×	×	×	×	×	✓	×	×
	[64]		2017	Y	Y	77	✓	×	✓	×	✓	✓	✓	×	✓	×	×	×	×	×	×	×	×
	[65]	DrThis	2018	Y	Y	41	✓	×	×	×	×	×	✓	×	×	×	×	×	×	×	✓	×	×
	[95]		2017	Y	Y	16	✓	×	✓	×	×	×	✓	×	△	×	×	×	×	×	×	×	×
	[36]		2017	Y	N	90	△	×	×	×	✓	✓	×	×	×	×	×	×	×	×	×	×	×
	[35]	RansomProber	2017	Y	Y	67	✓	△	✓	✓	✓	✓	✓	✓	✓	×	×	×	×	×	✓	✓	△
	[157]	Rands	2019	Y	Y	0	✓	×	✓	×	×	×	✓	×	×	×	×	×	×	×	×	×	×
	[37]		2019	Y	Y	3	✓	×	×	×	✓	✓	✓	×	△	×	×	×	×	×	×	×	×
	[14]		2020	Y	Y	1	✓	×	✓	×	✓	✓	✓	×	△	×	×	×	×	×	×	×	×
	[3]		2020	Y	Y	0	✓	×	✓	✓	✓	✓	✓	×	△	×	×	×	×	×	×	×	×
Combination of Static and Dynamic Analysis	[151]		2015	Y	Y	71	△	×	×	×	✓	✓	×	×	△	×	×	×	×	×	×	×	×
	[57]	DNA-Droid	2017	Y	Y	27	✓	×	✓	✓	×	×	✓	×	△	×	×	×	×	×	×	×	×
	[60]	RasHunt	2017	Y	Y	14	✓	×	✓	×	×	×	✓	×	△	×	×	×	△	×	×	×	×
	[134]	RansomWall	2018	Y	Y	28	✓	×	✓	×	✓	×	✓	×	✓	×	×	×	×	×	×	×	×
	[52]		2018	Y	Y	23	✓	×	✓	✓	✓	✓	✓	✓	△	×	×	×	×	×	×	×	×
	[75]	VoterChoice	2019	Y	Y	0	✓	×	×	×	✓	✓	✓	✓	△	×	×	×	×	×	✓	×	×
Honeypot Files	[113]		2016	N	Y	90	×	×	×	×	✓	✓	×	×	△	×	×	△	✓	✓	✓	×	×
	[50]		2017	N	Y	17	△	×	×	×	✓	✓	×	✓	✓	×	×	△	×	×	×	✓	×
	[58]	R-Locker	2018	N	Y	52	✓	×	×	×	✓	△	×	✓	△	×	×	△	✓	✓	✓	×	×

(Y: Yes; N: No; ✓: available; ×: unavailable; △: partially available).

sandboxed environments, by creating artificial sandboxed environments to allow ransomware to execute and to observe its behaviors. The approach of sandboxing in ransomware detection can suffer one or more of the following issues: (1) they are often resource-intensive, consuming higher CPU and memory usage, often making them not suitable as endpoint products for daily use, and not suitable on mobile devices with limited computing power and battery capacity; (2) analysis at the professional level is required, often too complicated for unsophisticated users; and (3) ransomware could implement virtualization-awareness or perform VM escape to evade sandboxing-based detection.

Static Analysis. (Table 5): *Static Analysis*, also known as *Static Code Analysis*, is the method to examine the source code or compiled executable files to search for the presence of malware, before the executable files are executed [73, 115]. Many studies ([10, 13, 15, 16, 39, 45, 48, 73, 74, 79, 98, 107, 127, 138, 154, 155]) performed *static analysis* for ransomware detection, and all claimed satisfactory detection results. However, those studies performing *static analysis* suffered from one or more of the following issues: (1) not all ransomware variants have the actual executable files present in the file system available for static analysis; (2) static analysis does not take into consideration the runtime variables, such as the parameters of API calls; (3) some assumed that future ransomware variants would all bear high similarity to existing ones, and did not consider encrypted, encoded, or dynamically loaded malicious code; (4) analyses of APIs or bycodes are often highly platform-specific, limiting their generalizability; and (5) ransomware could insert other benign code between malicious codes to obscure prominent static features.

App Behavioral Analysis. (Table 5): *App Behavioral Analysis*, also known as *Dynamic Analysis*, is the method to examine application behavior during its runtime, which includes a range of behaviors that are not limited to file encryption [140]. Some studies ([1, 3, 14, 18, 35–37, 64, 65, 95, 137, 157]) performed *app behavioral analysis* for ransomware detection with machine learning. Studies performing *app behavioral analysis* suffered from one or more of the following issues: (1) analyses of APIs are often platform-specific, limiting their generalizability; (2) statistical analysis of *app behavioral analysis* requires careful determination of statistical thresholds, but ransomware could still operate below the upper thresholds to evade detection; (3) *app behavioral analysis* can be easily bypassed by executing ransomware in containers (e.g., PowerShell runtime or VMs); (4) there are infinite combinations of operations of applications to user files, which requires proper profiling and classification; and (5) *app behavioral analysis* often takes time to be performed to achieve satisfactory accuracy, when terminating ransomware at the earliest opportunity is often required to minimize damage to user files and resources.

Combination of Static and Dynamic Analysis. (Table 5): A few studies ([52, 57, 60, 75, 134, 151]) combined both static and dynamic analysis. Studies combining *static analysis* and *dynamic analysis* may not only suffer from the issues aforementioned, but also need to address any conflicts if static analysis and dynamic analysis reach different conclusions.

5.1.4 User Files and Data. At the level of *User Files and Data* (Table 5), it is possible to deploy honeypot files to detect file modifications by ransomware [50, 58, 113]. Honeypot-based ransomware detection methods could catch ransomware attacks in progress, but could suffer three issues: (1) No deployment strategy could guarantee ransomware would attack honeypot files before attacking actual user files. Ransomware could in theory attack the most recently created or modified files by users first to avoid encountering honeypot files too early. (2) Users or other legitimate programs (e.g., compression utilities) could operate on the files and result in either false alarms or operational errors. (3) Honeypot files occupy disk space and could hinder other legitimate file operations; users cannot delete or move a folder if a honeypot file within a folder is in use. The issues aforementioned need to be addressed before honeypot-based ransomware detection methods could be considered more suitable for practical use.

5.1.5 Summary of Ransomware Detection. In summary, ransomware detection aims to discover the presence of either ransomware itself or its attack activities. From the ransomware CKC perspective, detection aims to proactively detect ransomware at or before the last (sixth) step of "actions on objectives." The main benefits of ransomware detection include the following: (1) it relies on characters or evidence of ransomware presence or consequences of ransomware damage, potentially reducing false-negative rates; (2) it can collect the OS context information to further improve detection accuracy; and (3) the detection features can be updated or learned by machine learning over time, further increasing detection accuracy. The downsides of ransomware detection include the following: (1) it is near the latter end of the CKC, when ransomware is already present in the OS; (2) it often does not involve users in the detection process; and (3) minimizing false-positive rates is always a challenge in ransomware detection.

Almost all of the ransomware detection proposals were engineering-based solutions, when the researchers attempted various methods, from discovering the presence of ransomware code, to detecting the progress of ransomware attacks. Those studies either passively used OS indicators and monitors, or actively inserted monitors into the OS to collect information on OS status and activities. Those studies all used different features of ransomware or characteristics of ransomware attacks; they may be effective against future ransomware variants only if those features or characteristics remain unchanged or similar. The studies used different ransomware and benignware samples, and withheld their complete engineering implementation source code, making it impossible to reproduce or thoroughly examine their results. While most compared their detection results to those of other detection solutions, they were not leveraging the same ransomware and benignware datasets. Therefore, the fairness of such comparisons and the validity of their self-claimed superiority could be questionable.

5.2 Ransomware Defense

This subsection lists and compares the studies to defend against ransomware while it is attacking its targets or victims (Table 6).

5.2.1 Hardware Level. Ransomware defense solutions at the hardware level primarily revolve around using hardware-based features to provide file or data recovery should ransomware attack. In [139], RDS3 used the file system stealthy spare space to keep the backup data of files. Four studies ([22, 68, 111, 147]) all utilized features of ***Solid State Drive*** (**SSD**) to implement data recovery in the firmware of SSD storage devices, which, unlike magnetic storage devices, do not overwrite original data when modifying, but write the data elsewhere before erasing blocks of original data, equivalent to the copy-on-write mechanism. Such hardware-based ransomware defense solutions can become OS-independent, and are more likely to deter against ransomware-like access patterns to storage devices at a lower level even if ransomware has gained the highest administrative privileges in the OS. However, such solutions are likely to suffer these inadequacies: (1) they could misjudge what are the last known "good" versions of user files; (2) it is difficult if ever possible to collect additional semantic information (e.g., type of file operations or state of the OS) at the storage level to assist in better decision making; (3) the storage overhead for frequently modified large files could be disproportionately large; (4) on SSD in particular, it could potentially interfere with the TRIM functionality, which is designed to eliminate unnecessarily copying invalid or discarded data pages, and thus affect the burst *write* speed and the overall performance; and (5) firmware-based ransomware defense utilizing SSD-specific copy-on-write characters will not work for other types of storage media, such as the magnetic HDD still dominant in large data centers.

5.2.2 User Mode. Ransomware defense solutions in the *User Mode* ([23, 47, 55, 56, 87, 92, 119, 126]) mostly focused on intercepting or backing up copies of encryption keys in case they were

Table 6. Comparison of Different Studies on Ransomware Defense

Research	Project Name	Year	Relevance			Evaluation				Output				Versatility						Strengths		
			Machine Learning Based	Peer Reviewed	Cited by Others	with Known Ransomware	with Unknown Ransomware	with Benign Application	against Competitors	Malicious Features Specified	Causal Relationships Analyzed	Accuracy Discussed	Efficiency Discussed	User Files Protected	Boot Sector Protected	Data Exfiltration Prevented	Effective Against Other Vectors	Cross Platform	Architecture Independent	Self Disclosure of Limitations	Minimal Delay until Full Effect	Difficult to Evade
[90]	CloudRPS	2016	Y	Y	49	✓	×	✓	✓	✓	✓	×	×	✓	×	×	×	×	×	×	△	×
[150]		2016	N	Y	24	✓	×	×	×	✓	✓	×	×	×	×	×	×	×	×	△	×	×
[119]		2017	N	Y	34	✓	×	✓	×	✓	✓	✓	×	✓	×	×	×	×	×	×	✓	×
[87]	PayBreak	2017	N	Y	124	✓	×	✓	×	✓	✓	✓	✓	✓	×	×	×	×	×	✓	✓	×
[92]		2017	N	Y	15	×	△	×	△	✓	✓	×	✓	✓	×	×	×	×	×	△	✓	×
[68]	FlashGuard	2017	N	Y	38	✓	×	✓	×	✓	✓	×	×	✓	×	×	✓	✓	✓	×	△	×
[139]	RDS3	2018	N	Y	8	×	×	✓	×	✓	×	×	×	△	×	×	△	△	×	△	×	×
[71]		2018	N	Y	4	×	×	×	×	✓	×	×	×	△	×	×	△	△	×	×	×	×
[55]	UShallNotPass	2018	N	Y	14	✓	×	✓	×	✓	✓	✓	✓	✓	×	×	△	×	×	✓	✓	×
[22]	SSD-Insider	2018	N	Y	24	✓	×	✓	×	✓	✓	✓	△	✓	×	×	△	✓	✓	×	△	×
[61]		2018	N	Y	1	✓	×	✓	×	✓	✓	△	×	×	✓	×	×	✓	✓	✓	×	×
[111]	Amoeba	2018	N	Y	8	×	△	×	✓	✓	✓	✓	✓	✓	×	×	✓	✓	✓	×	△	×
[147]	MimosaFTL	2019	N	Y	2	✓	×	✓	×	✓	✓	△	✓	✓	×	×	✓	✓	✓	×	△	×
[34]		2019	N	Y	0	×	×	×	×	✓	✓	×	×	✓	×	×	✓	✓	✓	×	×	✓
[56]	NoCry	2020	N	Y	3	✓	×	✓	✓	✓	✓	✓	✓	✓	×	×	×	×	×	✓	✓	×
[23]		2020	N	Y	4	✓	×	×	△	✓	✓	×	×	△	×	×	×	✓	×	×	×	×
[47]		2020	N	Y	0	✓	×	×	×	✓	✓	×	×	△	×	×	△	✓	×	×	×	×
[105]	Fitico	2020	N	Y	0	△	×	✓	×	✓	✓	×	✓	✓	×	×	✓	✓	×	✓	✓	△
[126]		2020	N	Y	0	×	×	×	×	✓	✓	×	×	△	×	×	×	×	×	×	✓	×

(Y: Yes; N: No; ✓: available; ×: unavailable; △: partially available).

later required to decrypt files encrypted by ransomware. The methods of intercepting, extracting, or recovering encryption keys of ransomware may help defend against some ransomware attacks, but such solutions suffer several issues: (1) they are platform-dependent or architecture-specific; (2) ransomware could bypass such defense by implementing its own random number generation and encryption key generation; (3) too many applications, including benign OS modules, may utilize those functionalities, and the ransomware defense implementations may have to implement application whitelisting on benign modules and accommodate for frequent module updates; (4) extracting encryption keys and keeping spare copies could itself introduce further security breaches if the key escrow is not sufficiently secure itself. Therefore, the strategy to attack ransomware key encryption could have limited practical usage. [105] proposed to enforce situation-aware access control to build malware-resilient file systems, with one possible application as defending user files against encryption by ransomware. While [105] prioritized early termination of ransomware attacks, their implementation was a prototype, and depended on future developments of reliable file validation methods to protect the file types of interest.

5.2.3 *User Files and Data.*

Ransomware *defense* solutions at the level of *User Files and Data* are primarily about file backups via secure methods ([71]) or via cloud storage ([34, 61, 90]). Cloud storage based or secure user file backups may defend against ransomware attacks by keeping versioned backup copies that are isolated from local copies, which can be easily attacked by ransomware. However, doing so requires sufficient network bandwidth and possibly redesign of local file systems to block further modifications until file upload to the cloud is completed. The network connection becomes essential to the success of such cloud solutions, and could become the single point of failure during network outages. Storing personal files on cloud storage could itself introduce further issues such as privacy, data safety, and version synchronization. Therefore, this defense strategy is possibly more suitable to protect smaller sized files that are stored on systems with good network connections and less frequently changed. At the level of *User Settings*, in [150], Weckstén et al. proposed to simply rename the Windows system tool "vssadmin.exe" that managed volume shadow copies, to defend against some Windows-based ransomware that executed "vssadmin.exe" commands, but their proposal was Windows-specific. The volume shadow copy service is not active on all computers, and there is no guarantee the latest most up-to-date copies of user files will be preserved.

5.2.4 *Summary of Ransomware Defense.*

To summarize, ransomware defense seeks to deter or intercept ransomware attacks in progress, or to revert damages by ransomware, without necessarily detecting ransomware. From the CKC perspective, ransomware defense tends to occur around the same time as ransomware detection, but takes a more reactive approach. The main advantages of ransomware defense include: (1) it provides a general protection and is often non-specific to particular ransomware variants; (2) it is resilient to ransomware evolution, as long as the attack vectors remain unchanged. However, the disadvantages of ransomware defense can include: (1) the defense method is often static and can be easily circumvented; (2) it must augment into the OS or change OS settings to be effective. The ransomware defense proposals could only be reactive to the presence of ransomware, not proactive nor preventative. There lacks a clear consensus on what constitutes successful defense against ransomware (*i.e.* prevention of ransomware execution or ransomware damages). Many ransomware defense proposals were engineering-based, and often required modifications of OS modules or settings. Each ransomware defense solution often focused on one aspect of the ransomware intrusion principles. Ransomware could attempt to mimic benignware or to change its intrusion principles to bypass existing ransomware defense.

5.3 Ransomware Prevention

This subsection lists and compares the studies to prevent ransomware from reaching its targets or victims (Table 7).

5.3.1 *Hardware Level.*

At the hardware level, ransomware *prevention* can be implemented as *firmware-based access control* ([5, 136]). Ransomware prevention at the hardware level often took an "all-or-none" approach and often could not adequately distinguish between legitimate usage or malicious usage, both of which could come from either the OS or user applications. The prevention methods were not sufficiently flexible to accommodate the dynamic user environment or to consider different use cases. There could exist security loopholes where malware could still bypass the prevention methods to access files via calling OS modules. Therefore, this prevention strategy is possibly more suitable to protect systems with restricted functionalities or use cases running only fully trusted applications.

5.3.2 *User Files and Data.*

At the level of *Physical Files and Data, Settings*, some studies ([84, 145]) advocated at the user application level to enforce application whitelisting. Application

Table 7. Comparison of Different Studies on Ransomware Prevention

Research	Project Name	Year	Relevance			Evaluation				Output				Versatility						Strengths		
			Machine Learning Based	Peer Reviewed	Cited by Others	with Known Ransomware	with Unknown Ransomware	with Benign Application	against Competitors	Malicious Features Specified	Causal Relationships Analyzed	Accuracy Discussed	Efficiency Discussed	User Files Protected	Boot Sector Protected	Data Exfiltration Prevented	Effective Against Other Vectors	Cross Platform	Architecture Independent	Self Disclosure of Limitations	Minimal Delay until Full Effect	Difficult to Evade
[97]		2007	N	Y	114	×	×	×	×	×	×	×	×	△	△	△	×	✓	✓	×	×	×
[136]		2017	N	Y	2	△	×	×	×	✓	✓	×	×	×	✓	×	✓	✓	×	×	✓	✓
[145]		2018	N	Y	4	✓	×	✓	×	✓	✓	×	✓	△	△	△	×	×	×	×	✓	×
[17]	AntiBotics	2018	N	Y	6	✓	×	✓	×	✓	✓	✓	×	✓	×	×	×	×	×	✓	✓	×
[143]		2018	N	Y	32	×	×	×	×	✓	✓	×	×	△	△	△	×	✓	✓	✓	×	×
[70]		2019	N	Y	2	×	×	×	×	✓	✓	×	×	△	△	△	△	✓	✓	×	×	△
[69]		2019	N	Y	20	✓	×	×	×	✓	✓	×	×	△	△	△	△	✓	✓	✓	×	△
[5]	Key-SSD	2019	N	N	0	△	×	✓	×	×	×	×	×	✓	✓	×	×	✓	✓	×	✓	△
[93]		2019	N	Y	5	✓	×	✓	×	✓	✓	✓	×	✓	×	△	△	×	×	✓	✓	×
[84]		2020	N	Y	1	✓	×	×	×	×	×	×	×	✓	×	△	×	×	×	×	✓	×
[26]		2020	N	Y	0	×	×	×	×	×	×	×	×	△	△	△	×	✓	✓	×	×	△

(Y: Yes; N: No; ✓: available; ×: unavailable; △: partially available).

whitelisting may be able to mitigate some ransomware attacks by preventing its execution. However, it may not be able to fully prevent all execution paths of ransomware, and may not prevent ransomware from either attacking via whitelisted applications or masquerading as those applications. The implementations of application whitelisting can require changing OS settings or embedding architecture-specific modules into OS, complicating the matter and limiting its practicality.

At the level of *File and Data Maintenance*, ransomware prevention could be implemented as file-level access control ([17, 93]), and protected folders as part of the antivirus products (e.g., BitDefender, Trend Micro and Microsoft Defender). Access control at the file level may disrupt the attacks of some ransomware types, but they all appeared to require users to adapt to different ways of interacting with the OS, may not fully prevent ransomware from attacking user files via permitted OS modules, and may not prevent ransomware masquerading as benign programs. Therefore, existing file level access control may offer limited prevention against ransomware attacks.

5.3.3 *Users, System Admins and Organizations.*

At the level of *Users, System Admins and Organizations*, ransomware *prevention* can be implemented as user education and awareness ([26, 97, 143]), or company policies on SecureOps ([69, 70]). Those ransomware prevention methods are all user-centric as proposed in [69, 97, 143]. They recognized that users could be a weak link in the process of preventing ransomware attacks, but recommendations are often based on industry experience without being properly evaluated. Interventions at this level could add another valuable layer in mitigating ransomware, but are probably insufficient as standalone ransomware mitigation methods.

5.3.4 Summary of Ransomware Prevention. In summary, ransomware prevention aims to prevent ransomware from reaching the victim OSs in the first place. From the CKC perspective, it tends to sit at the second (delivery), third (exploitation) or fourth (installation) step. By acting during the upper steps of the CKC, it has the potential to filter out some ransomware variants to prevent them from reaching target OSs. However, the effectiveness of ransomware prevention is often difficult and sometimes impossible to measure or estimate. Ransomware prevention can be achieved via hardware-based physical isolation, implementation of OS settings or group policies to deter ransomware proliferation, and improvements of user awareness and organizational procedures to minimize the risk of ransomware infiltrations. The ransomware prevention proposals surveyed included both engineering-based and observational studies. The engineering-based solutions could suffer the same issues aforementioned. The observational studies involved several objective human factors, such as effectiveness of user awareness training and adherence to corporate IT policies, the effectiveness of which can be difficult to quantify. Some of such studies also made recommendations on how to improve the security situation of target OSs, but the recommendations were based on the industry experience of some cybersecurity professionals. The effectiveness of observational studies could only be estimated through qualitative analysis, due to the methodologies employed by them and the difficulty of implementing controlled trials.

6 SURVEY INSIGHTS

Based on our review of research articles in Section 5, we have made the following observations and summarized them as follows.

6.1 Major Themes Identified

We have identified three groups of major themes among the existing anti-ransomware proposals.

6.1.1 Programmatic, Data-Centric or User-Centric Approach. The majority of the anti-ransomware studies took one of the three approaches (programmatic, data-centric or user-centric). Some studies that took the *programmatic* approach only focused on specific features of the code (e.g., APIs called, OpCode sequences, random number generation), either static or dynamic, without adequately considering the object (e.g., user files, OS settings) of code executions; many such studies performed static or dynamic analysis. Such proposals either assumed or concluded that certain programmatic operations are more risky or harmful than others. A *programmatic* approach can be problematic when it fails to realize that the result of code execution not only depends on the programming logic, but also the data it processes. Some proposals that took the *data-centric* approach made decisions based on a broad set of sources, taking into consideration the objects of code executions (mainly changes in user files, but also including network traffic and OS indicators of compromise). A *data-centric* approach facilitates more flexibility of making decisions on ransomware mitigation, but can also be taken advantage of, when ransomware could manipulate its behaviors to mimic known benign code. Still, neither the *programmatic* nor the *data-centric* approach considers the factors of users, user operations and user decisions in mitigating ransomware, which are the focus of a *user-centric* approach. Users, whether experienced or unsophisticated, are often the data owners or data custodians of the files and data that are attack targets of ransomware; they execute applications and can notice OS abnormalities, and respond to ransom messages if displayed by ransomware. Most *user-centric* approaches have attempted to apply security disciplines of operating OSs to users (e.g., SecureOps guidelines and procedures) or to promote awareness in ransomware prevention (e.g., via user education and training). However, the non-homogeneity of user experience, expectations and responsiveness to either ransomware awareness or security disciplines can add to the complexity of fine-tuning *user-centric* ant-ransomware approaches.

6.1.2 Process-Oriented or Outcome-Oriented. In the context of anti-ransomware proposals, an *outcome-oriented* approach encourages a focus on the end state (i.e., whether something is likely to be ransomware or not) the researchers aim to achieve, while a *process-oriented* approach involves elaboration on the step-by-step analytical process (i.e., why something is likely to be deemed ransomware) that leads to the aforementioned end state. The peak of the number of anti-ransomware proposals around 2018 (as seen in Figure 1) corresponded to the sudden surge in research interest in machine learning, and its subsequent application on available ransomware datasets (either self-constructed or shared). A significant proportion of those ML-based proposals (e.g., [4, 7, 9, 10, 15, 16, 20, 21, 44, 45, 57, 60, 65, 74, 79, 86, 95, 98, 99, 110, 112, 127, 129, 141, 154, 155, 157]) were purely *outcome-oriented* and did not specify the malicious features found by their ML algorithms among ransomware samples, whereas a few other ML-based proposals (e.g., [13, 48, 51, 63, 78, 94, 124, 131, 132, 144, 146, 158]) specified the malicious features they detected or concluded, but did not analyze whether there had been a causal relationship between each feature they considered malicious and the actual ransomware attack mechanism. Relying on ML to make the decision of ransomware classification without further analysis or verification of the results presented by ML can cause several issues: the ML algorithms can pick up features irrelevant to ransomware attacks, can overfit to the training samples, and do not adequately contribute to the theoretical understanding of ransomware attack mechanisms. It is possible that different combinations of ML algorithms, ransomware datasets, and testing environments can result in more of such papers published, but their external validity (i.e., actual contribution to mitigating newer ransomware variants) should be further scrutinized. The *process-oriented* studies that have explained how they have reached their conclusions, however, enable reviewers and readers to better understand their analytical process to make better judgments on the external validity of such studies.

6.1.3 Programming Freedom vs. Programming Restrictions. Two distinctive approaches have been noticed among the existing anti-ransomware proposals: *programming freedom* vs. *programming restrictions*. The notion of *programming freedom* promotes the free choice of writing any programming code, when code can perform all types of operations on user files and data, and is presumed benign until deemed to be malicious (ransomware-like) by anti-ransomware research. Many ML-based proposals belong to this category and only classify the piece of code as ransomware if it exhibits features that resemble those found in previously known ransomware samples. For example, some studies examined whether there were ransomware-like encryption activities ([8, 62, 85, 86]) or ransomware-like file system activities ([20, 63, 66, 91, 130]). However, promoting *programming freedom* means researchers have to analyze an infinite number of possible combinations of benign or malicious programming operations, which enables flexibility but can be a tedious task. The notion of *programming restrictions* preemptively emphasizes restrictions on what operations might be considered malicious; codes performing such operations are considered malicious until proven benign. Proposals that promote *programming restrictions* are often rule-based. For example, some studies ([23, 47, 55, 56, 87, 92, 119]) intercepted encryption key or random number generations, operations they considered dangerous and ransomware-like. Promoting *programming restrictions* can make the proposals arbitrary, subject to the bias of researchers, and can risk compromising the flexibility to meet the ever changing user demand on OS functionalities.

6.2 Limitations of Existing Anti-Ransomware Proposals

The existing anti-ransomware studies have contributed to the knowledge of ransomware, but have nevertheless suffered one or more of the following issues.

6.2.1 High Accuracy Claimed Despite Wildly Different Methodologies. The most pressing issue we have noticed is that, among the studies that have claimed accuracy of their detection results, almost all of them claimed very high accuracy, despite their wildly different methodologies. A more comprehensive evaluation should ideally exhibit a high true-positive rate on a range of known and unknown ransomware, and a low false-negative rate on a variety of benign applications [121]. The majority of studies we surveyed claimed ransomware detection accuracy of more than 95%. Some studies (e.g., [2, 32, 81, 114]) even claimed 100% detection rates, whereas a few others (e.g., [58, 73, 113]) did not discuss the detection accuracy. The way those studies presented their accuracy also varied. Some studies (e.g., [146, 158]) only evaluated with known ransomware samples without testing with unknown ransomware or benign applications. A few others (e.g., [62, 66, 72, 83, 102]) only tested with one or very few known ransomware samples. The established validity and reliability of an anti-ransomware study, based on selected ransomware samples, does not necessarily mean the same study is valid and reliable for all ransomware samples; it is valid and reliable only for those ransomware samples that are very similar to the ones on which the proposal's validity and reliability have been established. Without comparing different studies on the same ransomware samples, claims of superiority by some studies simply based on high accuracy may not be fully justified. Meanwhile, we acknowledge that the dilemma encountered when evaluating anti-ransomware studies was not only caused by the issue itself, but also by the way researchers presented their results. We suggest readers and reviewers of such studies should put the detection results reported by the studies into perspective, considering the quality and quantity of ransomware and benign samples collected and evaluated by them.

6.2.2 Evaluation With Different Ransomware Samples. The evaluations of anti-ransomware proposals have usually been performed using sets of ransomware-related data that have been prepared locally by the researchers. Apart from one study (e.g., [73]) that did not present its evaluation with ransomware, many others have chosen to obtain their experiment data from one or more of the following sources: ransomware samples downloaded from VirusTotal or other ransomware deposits, files downloaded by crawling online forums of ransomware discussions, datasets of ransomware attack features (e.g., event logs or network traffic logs), or simulating ransomware-like activities. Evaluating with locally prepared ransomware-related data is often problematic itself, because (1) researchers may be unable to obtain a balanced variety of different ransomware samples, (2) it often did not consider whether the ransomware sample itself was still active or not; (3) the classification of ransomware and malware in general often lacked universal standards, when different antivirus vendors on VirusTotal may choose to classify the sample differently, disagree on whether a sample is malicious or not, or mark one module of known ransomware as the ransomware itself; (4) in the scenarios of hacking to deploy ransomware, or fileless ransomware scripts, no malicious executable files can be obtained; and (5) automating the process of testing each ransomware sample in controlled environments before reverting the environments to the pretesting stage could be a tedious task. The aforementioned issues were also shared by various studies that were evaluated with ransomware samples. It is a possible future research direction to propose a reliable and automated way of collecting, classifying, and testing active ransomware samples.

6.2.3 Lack of Investigation of Malicious Features, or Insufficient Justification of the Selection of Malicious Features. A comprehensive ransomware mitigation research should ideally investigate and disclose the malicious features used for its ransomware mitigation, and discuss the justification of the selection of such features. Doing so enables other researchers to better understand the ransomware attack mechanisms, and to examine whether the features selected are appropriate and justified. While many studies (e.g., [1–3, 8, 11, 12, 14, 17, 22, 23, 28, 32, 34–40, 43, 47, 50, 52, 55, 56, 58, 61, 62, 64, 68–70, 72, 73, 75, 80, 81, 83, 87, 90–93, 101, 102, 105, 107, 108, 111, 113, 114, 117–119, 125, 126, 130, 133, 136, 138, 142, 143, 145, 147–151]) both disclosed the malicious features of ransomware they found and justified their selection, some studies (e.g., [13, 48, 51, 63, 71, 78, 85, 94, 124, 131, 132, 134, 139, 144, 146, 158]) only disclosed ransomware malicious features they selected without justification, and others (e.g., [4, 5, 7, 9, 10, 15, 16, 18, 20, 21, 26, 44, 45, 57, 60, 65, 74, 79, 84, 86, 95, 98, 99, 110, 112, 127, 129, 141, 154, 155, 157]) did not discuss ransomware malicious features at all. Ransomware malicious features are by design, and ransomware developers can implement different features in different attack vectors or patterns. It is possible that some studies relied on machine learning to perform the data-mining work of feature extraction or clustering. However, without further examining whether the results of data mining made sense, and whether the relationship between their chosen malicious features and the ransomware attack mechanism was casual or causal, such studies ran the risk of over-fitting to their samples.

6.2.4 Lack of Prioritization of Early Termination of Ransomware. Ransomware mitigation strategies should prioritize early termination of ransomware activities. The majority of the studies we surveyed did not prioritize this or did not discuss how soon they could terminate ransomware attacks; some studies even suggested extremely long observation periods (e.g., 30 days by [158]) to obtain the maximum detection accuracy. Although ransomware is considered malware, ransomware attacks are different from traditional malware attacks in that they could result in irreversible damage (i.e., encrypted files, damaged MBR, or data exfiltration). Previously, when unsure whether a suspicious process was malware or not, it was possible to let it execute in the OS while observing its behavior to accumulate more evidence, required to obtain a more accurate malware classification result. However, this cannot always apply to behavioral analysis of ransomware. Ransomware can aggressively encrypt user files within a very short period of time, damage MBR in one *write* operation, or exfiltrate user data without writing to the file systems. Therefore, ransomware mitigation requires early termination of ransomware activities, sometimes at the cost of the luxury of having extended lengths of time to perform proper behavioral analysis. Studies on ransomware mitigation should also report whether they are capable of early termination of ransomware attacks.

6.2.5 Incomplete Solution to Protect OS from Multiple Attack Vectors. While some earlier studies attempted to deter UI-locking behaviors of some earlier variants of ransomware, recent studies almost universally aimed to detect, defend against, or prevent the encryption behavior of ransomware, and many assumed that ransomware was only in the form of either executable files on desktop platforms or apk files on Android. The recent emergence of fileless ransomware scripts, VM-based ransomware, or manually deployed ransomware by hacking can easily make analysis on executable files (e.g., static analysis or API analysis) irrelevant. Ransomware attacking MBR leaves proposals monitoring modifications of user files (e.g., file access control, backup using disk spare space) ineffective. If data exfiltration based ransomware does not perform encryption activities, it will not be captured by anti-encryption-based methods (e.g., encryption key interception, encryption detection). Therefore, a more complete anti-ransomware solution should attempt to protect the OS from multiple attack vectors if possible, and should be difficult to circumvent.

6.2.6 Lack of User Involvement in Decision Making. The majority of the published studies we surveyed took either a program-centric approach (e.g., static analysis, API analysis) or a data-centric approach (e.g., file system analysis, network activity analysis), without considering user involvement in decision making. Many of them will be unable to distinguish between user-initiated benign encryption and unknown-ransomware-driven malicious encryption, or between user-driven data transfer and malicious data exfiltration. Some studies chose to involve users by displaying dialog boxes to ask the user to decide whether to permit the execution of the process or some of its operations. However, such an approach can sometimes be problematic, because it could quickly result in alarm fatigue by the users, and unsophisticated users cannot fully consent without adequate understanding of the process or its pending behaviors [49, 96]. User involvement is considered essential in fighting the modern war against malware, especially ransomware, because it can provide valuable information to help security software examine whether the true intent of the process of interest aligns with users' expectations, when malware often performs damages against them [128, 135]. However, involving the user in decision making to combat ransomware has its own additional challenges. Apart from the aforementioned issue of user content, there are often limited ways of user interacting with the OS, and there lacks a unified approach in implementing user-driven access control [128].

6.2.7 Lack of Self-Disclosed Limitations of Research. The armed race against the ever evolving ransomware is unlikely to end in the near future, and requires researchers to constantly get updated with the latest threat intelligence and to refine the mitigation strategies. The *limitation* of a study is the systematic bias that was not or could not be controlled by researchers and could affect the study results inappropriately [123]. The researchers may have designed their study for a particular group of ransomware, a particular attack vector, or some other attribute that would limit to which ransomware their findings were applicable. Because no research is perfect, existing research should consider including an adequate discussion of its limitations in its submitted manuscript and suggest future research directions. Such an adequate coverage of research limitations should include discussions of both internal validity (accurately measures what it intends to measure) and external validity (the sample results accurately represent the results of the entire target samples) [123].

Among the ransomware mitigation studies we surveyed, almost all of them claimed superiority over existing research, yet 63% of the studies (74 out of 118) did not include any limitations of their research in their manuscripts, whereas 5% of them (6 out of 118) gave short shrift to the limitations of their studies. This prompted concern about whether they have adequately appreciated the limitations of their research, have decided not to point out their limitations, or have left the job of making note of limitations to manuscript readers and reviewers [123]. We suggest that anti-ransomware researchers should include sufficient discussions of their research limitations, to enable readers to place an appropriate level of credit on the findings of the study as warranted. We also advise that readers should be skeptical of both the internal and external validity of published research, and consider repeating the same research design on different ransomware samples and attack vectors.

6.3 Discussion

The ransomware landscape has evolved from simpler hoax-ware and screen lockers, to worm-like ransomware, manually deployed encryption, and now threats of data breach. With the continuing proliferation of more networked devices and the advancement in encryption and obfuscation techniques toward more aggressive and evasive ransomware variants, the incidences of ransomware attacks will continue to occur. It has become more essential than ever before to secure personal data and devices of all platforms against ransomware attacks.

6.3.1 The Impact of Darknet in Ransomware Evolution. Darknet, an unregulated area of the Internet, has worsened the ransomware pandemic by laundering ransom payment via cryptocurrencies, offering ***Ransomware-as-a-Service*** (**RaaS**), and recruiting cybercriminals to launch sophisticated ransomware attacks [109]. Although earlier versions of ransomware (e.g., some screen lockers) often demanded payments using telegraph transfer, bank checks, and gift cards,

Table 8. Possible Ransomware Countermeasures Against Existing Ransomware Mitigation Strategies

Existing Ransomware Mitigation Strategies		Countermeasures by Ransomware	
		Ransomware Obfuscation	**Ransomware Circumvention**
Detection	API or Opcode Analysis	Code Obfuscation	Fileless ransomware [76, 100] VM-based ransomware (e.g., *Ragnar*)
	App Behavioral Analysis	Behavioral Obfuscation	
	Encryption Behavior Analysis	Use Different Encryption Library	
	File System Activity Analysis	Obfuscate File System Activities Manipulate File Entropy Values [103]	Direct Disk Access
	Detection of OS Compromise	Behavioral Obfuscation	VM-based ransomware (e.g., *Ragnar*) Perform Data Exfiltration [6]
	Hardware-Level Detection		Direct Disk Access Overloading Hardware [43]
	Network Activity Analysis	Network Traffic Obfuscation	No Network Traffic [27]
	OS Kernel Activities	Behavioral Obfuscation	Compromising Boot Sector (*Petya*) Perform Data Exfiltration [6]
	Sandboxing	Detection of Sandboxing or Virtualization	Virtual Machine Escape
Defense	Cloud Storage and Backup	Deletion of Backup Copies Obstruction of Backups	Disconnection of Network
	Encryption Interception	Usage of Different Encryption Library	VM-based ransomware Perform Data Exfiltration [6]
	OS Hardening	Social Engineering [53]	Targeted Vulnerability Exploitation [46]
Prevention	Application Whitelisting	Masquerading as Whitelisted Applications [104]	Attacks via Whitelisted OS Modules
	File or Firmware-Based Access Control	Obfuscate File Access Patterns [103]	
	SecureOps	Behavioral Obfuscation	Hacking to Deploy Ransomware [41, 148]
	User Education and Awareness	Social Engineering [53]	

recent ransomware almost universally required ransom payments via cryptocurrencies, due to the anonymity and irreversibility of cryptocurrency transactions [42, 109]. Various money laundering schemes have been offered via darknet, enabling cybercriminals to channel through their illegal gains [109]. During earlier ransomware attacks, the ransomware developers were often the attackers themselves; the emergence of RaaS enabled three key economic tiers (ransomware developers, RaaS, and distributors) to divide their tasks and share the illegal profits in the ransomware supply chain [109]. RaaS made it possible for ransomware developers to rapidly develop newer ransomware variants to evade antivirus detection, whereas the ransomware distributors could focus on searching for larger but vulnerable targets to launch attacks of larger scales. Additionally, darknet provided the platform for cybercriminals to be recruited to perform ransomware attacks of every increasing sophistication, and to attack organizations anywhere in the world [42, 109]. The growing involvement of darknet in ransomware campaigns suggested that the corporation of law enforcement against illegal activities on darknet may assist in deterring ransomware attacks.

6.3.2 Possible Ransomware Countermeasures Against Existing Ransomware Mitigation Proposals. In this subsection, we hypothesize how some of the existing ransomware mitigation strategies may become less effective (obfuscation) or even completely ineffective (circumvention) [77]. Ransomware has constantly been investigating new ways of defeating existing anti-ransomware measures. Since late 2019, a hybrid variety of ransomware attacks has emerged, when traditional ransomware techniques (i.e., file encryption) are combined with new techniques, such as data exfiltration. Here we explore and hypothesize some possible countermeasures by ransomware against existing ransomware mitigation proposals (Table 8), from an offensive point of view, and consider them as possible vulnerabilities. Some of those countermeasures have already been employed by very recent ransomware variants. We here only enlist them without evaluating the probability or impact of those possible ransomware countermeasures, which can be a subject of future research. Ransomware has evolved from one executable program to encrypt user files, to a type of cyber attack that will blackmail users with any valuable data.

6.3.3 The General Trend of Loss of Relevance of Engineering-Based Anti-Ransomware Research. We believe much of the engineering-based anti-ransomware research is gradually losing its importance or relevance. Ransomware research can include a very wide field of work that can embrace several different goals, which can include observational studies on ransomware features and behaviors (e.g., [36, 67, 80, 121, 152]), engineering solutions to detect or defend against ransomware (e.g., [43, 68, 124, 130]), and offensive techniques to hypothesize how to evade current ransomware mitigation (e.g., [31, 103, 120]). Expectedly, the majority of ransomware studies surveyed by us appeared to be engineering solutions to mitigate ransomware in controlled environments, as there was a pressing need to mitigate ransomware threats. However, many anti-ransomware engineering solutions attempted to become one-size-fits-all solutions, when those researchers often developed their hypotheses in *ad hoc* manners, without being informed by data from observational or offensive studies. Some engineering solutions (e.g., [4, 5, 7, 9, 10, 15, 16, 18, 20, 21, 26, 44, 45, 57, 60, 65, 74, 79, 84, 86, 95, 98, 99, 110, 112, 127, 129, 141, 154, 155, 157]) did not even attempt to define clear threat models, while many of them relied on machine learning algorithms to select differential features among the ransomware samples they investigated; such methods would only work if future ransomware samples exhibit identical or extremely similar features to the existing ones investigated in those studies. In malware research, a threat model is required to define what is to be protected and what methods will be leveraged for such tasks [29]. Clearly defined threat models enable ransomware researchers to appropriately position their work in the correct context, whereas the lack of such clear definitions can raise doubts on the relevance, viability, and limitations of such studies. Some anti-ransomware engineering solutions appeared to fail to consider real-world connections, with either excessively long detection time (e.g., [139, 157]) or complicated user involvement (e.g., [17, 66, 93]), making them less likely to be further adopted by home users or the antivirus industry. Another pitfall of engineering-based solutions was the lack of a balanced repository of ransomware and benign samples. Many studies appeared to exemplify their solutions via a limited unbalanced portfolio of ransomware samples they could collect, via Proof-of-Concept simulations, or simply via tabletop exercises. They relied on non-homogeneous antivirus labels of ransomware as ground-truth, assumed that crawled or well-known applications were benign without validations, used non-reproductive methods to define datasets and design evaluations, and simply compared their results to those of other studies without considering the differences in samples and environments. All the aforementioned pitfalls of engineering-based anti-ransomware solutions can affect the validity of such studies and/or their applications in the real world.

6.3.4 Proposed Ransomware Mitigation Framework. Since the primary attack principle of ransomware is to gain control of valuable user resources and to exclude users' access, ransomware mitigation is effectively an access control issue—to grant appropriate access of user resources to users' legitimate code but to deny access by ransomware. A key challenge in anti-ransomware research is to ensure the alignment of user expectations of the code behaviors on user resources, to the actual behaviors exhibited by the code. The user can choose to execute the code, enter data input to the code, or operate on the UI of applications. However, once the code is executed, the user often has neither the insights nor the controls of code behaviors. The user in operation may have expectations of how the user resources may be operated on, managed, or modified, but such expectations are not guaranteed. Ransomware often first has to gain the user's trust in executing on the user's OS, only to let the user come to the realization later that access to the user resources has been deviated from the expected state. We believe the fundamental cause of repeated and rampant ransomware episodes is the lack of appropriate access control in information access, both in data retrieval and data modification. We propose a layered approach to filter out as many malicious and invalid access operations as possible via prevention and defense, while relying on detection to detect the remaining malicious ones. This is in line with the general consensus of the "defense in depth" strategy: after prevention strategies prevent many threats from reaching the target, defense strategies defend the target against threats from taking actions on the target, and any remaining threats will be detected when their actions are observed.

The most critical dilemma to address the issue of access control while developing ransomware mitigation strategies is to preserve the freedom of users to achieve their purposes with the OS while restricting ransomware from its attacks or limiting its attack damage to a minimum. There are many possible approaches that have been proposed, but the better solutions should be ones that will not easily become outdated by the ever evolving ransomware landscape. However, to properly develop a mitigation framework toward the aforementioned research direction, the following additional issues also need to be addressed: (1) there should be further research to define what is acceptable and desirable access request to user files and resources, for the purpose of reading and modifying them; (2) how to properly engage users to make access control decisions; and (3) how to minimize the risk of overwhelming them with the burden of anti-ransomware defense.

6.4 Limitations of This Survey

Despite our best effort to survey as many relevant papers as possible, we present the limitations of this survey.

6.4.1 Availability and Quality of Existing Research. A fundamental constraint that was identified during our survey process was a general lack of literature that discussed the effectiveness of proactive countermeasures (e.g., ransomware prevention via system settings or user education). Articles covering such topics tend to merely present the concepts without performing controlled evaluation on the effectiveness. As a result, there is a lack of primary articles that compared and contrasted their effectiveness against that of reactive countermeasures (e.g., ransomware detection and defense). Additionally, there appears to be a lack of research into the latest ransomware evolution into fileless scripts, VM-hosted encryption, and ransomware deployment via hacking, designed to evade all existing analysis. Instead, we tentatively assess the current research on ransomware mitigation against those new variants, discuss their inadequacy in mitigating the new threats, and propose possible new research directions.

6.4.2 Challenges in Validating the Internal and External Validity of Some Studies. We have found it challenging in validating the validity of some studies simply based on their manuscripts. The internal validity of a study can be best examined when its research method can be scrutinized or repeated on different test samples. Some researchers only briefly described their research methods, or in some cases even avoided explaining it at all. Some implemented prototype software and made the source code available, while others withheld their software. Therefore, evaluating the internal validity of many studies became difficult if not impossible. The external validity of a study depends on how different the ransomware samples used by them deviate from the ransomware variants active in the real world. While many researchers downloaded as many ransomware samples as possible, subject to availability, some used only a few families of ransomware, used simulators, or even case studies; there has clearly been assumptions they have covered the required scenarios of ransomware features in the wild. To the best of our knowledge, none of the existing studies has tested with either real samples or simulation of VM-based ransomware or ransomware with data exfiltration, which puts the external validity of those studies in question.

7 CONCLUSION AND FUTURE WORK

We have performed a survey on existing ransomware mitigation proposals, and proposed a taxonomy in classifying the proposals based on where they operate within the general OS architecture. Unlike other surveys that often focused on ransomware detection or simply compared the methodologies and detection accuracy results reported by those proposals, we proposed a set of unified metrics in evaluating studies on ransomware mitigation, and surveyed 118 such studies, covering all aspects of ransomware mitigation—detection, defense, and prevention. We classified and grouped them based on their primary method of ransomware mitigation, and attempted to compare and contrast their strengths and weaknesses. We found that those proposals took a programmatic, data-centric, or user-centric approach, were either process-oriented or outcome-oriented, and promoted either programming freedom or advocating programming restrictions. Other important findings in this survey include the following: many ransomware mitigation proposals claimed high efficacy and superiority over other ones, but their validity cannot be easily validated; the issue is further complicated by the difficulty in obtaining working ransomware samples and the vastly different ways researchers evaluated their proposals and presented their results. We also observed that some studies lacked insight into the attack mechanisms of ransomware, or could be easily outdated by newer countermeasures by ransomware.

The fundamental issue of repeated ransomware attacks, whether it involves data encryption or data exfiltration, is the lack of appropriate access control to files and OS resources to ensure that code behaviors on how user files and data are assessed should be consistent with user intentions; it is a major research gap shown by the results of our survey. Ransomware will continue to evolve with new features, attack vectors, and improved attempts to lessen or completely bypass existing mitigation proposals. We expect more studies of different combinations of ransomware samples, feature selection, and mitigation strategies will continue to be produced by researchers, but any proposal that does not address the fundamental issue of access control can be easily outdated over time, when newer variants of ransomware obfuscates or abandons features known to anti-ransomware measures. Future research is required to provide adequate and appropriate access control that both enables normal user operations on the OS and hinders ransomware attacks.

REFERENCES

[1] Mohammad Mehdi Ahmadian and Hamid Reza Shahriari. 2016. 2entFOX: A framework for high survivable ransomwares detection. In *2016 13th International Iranian Society of Cryptology Conference on Information Security and Cryptology (ISCISC'16)*. IEEE, 79–84.

[2] Mohammad Mehdi Ahmadian, Hamid Reza Shahriari, and Seyed Mohammad Ghaffarian. 2015. Connection-monitor and connection-breaker: A novel approach for prevention and detection of high survivable ransomwares. In *2015 12th International Iranian Society of Cryptology Conference on Information Security and Cryptology (ISCISC'15)*. IEEE, 79–84.

[3] Yahye Abukar Ahmed, Barış Koçer, and Bander Ali Saleh Al-rimy. 2020. Automated analysis approach for the detection of high survivable ransomware. *KSII Transactions on Internet and Information Systems (TIIS)* 14, 5 (2020), 2236–2257.

[4] Yahye Abukar Ahmed, Barış Koçer, Shamsul Huda, Bander Ali Saleh Al-rimy, and Mohammad Mehedi Hassan. 2020. A system call refinement-based enhanced Minimum Redundancy Maximum Relevance method for ransomware early detection. *Journal of Network and Computer Applications* (2020), 102753.

[5] Jinwoo Ahn, Donggyu Park, Chang-Gyu Lee, Donghyun Min, Junghee Lee, Sungyong Park, Qian Chen, and Youngjae Kim. 2019. KEY-SSD: Access-control drive to protect files from ransomware attacks. https://arxiv.org/abs/1904.05012.

[6] Muna Al-Hawawreh, Frank den Hartog, and Elena Sitnikova. 2019. Targeted ransomware: A new cyber threat to edge system of brownfield industrial Internet of Things. *IEEE Internet of Things Journal* 6, 4 (2019), 7137–7151.

[7] Bander Ali Saleh Al-rimy, Mohd Aiziani Maarof, Mamoun Alazab, Fawaz Alsolami, Syed Zainudeen Mohd Shaid, Fuad A. Ghaleb, Tawfik Al-Hadhrami, and Abdullah Marish Ali. 2020. A pseudo feedback-based annotated TF-IDF technique for dynamic crypto-ransomware pre-encryption boundary delineation and features extraction. *IEEE Access* 8 (2020), 140586–140598.

[8] Bander Ali Saleh Al-rimy, Mohd Aizaini Maarof, Mamoun Alazab, Syed Zainudeen Mohd Shaid, Fuad A. Ghaleb, Abdulmohsen Almalawi, Abdullah Marish Ali, and Tawfik Al-Hadhrami. 2020. Redundancy coefficient gradual upweighting-based mutual information feature selection technique for crypto-ransomware early detection. *Future Generation Computer Systems* (2020).

[9] Bander Ali Saleh Al-rimy, Mohd Aizaini Maarof, Yuli Adam Prasetyo, Syed Zainudeen Mohd Shaid, and Asmawi Fadillah Mohd Ariffin. 2018. Zero-day aware decision fusion-based model for crypto-ransomware early detection. *International Journal of Integrated Engineering* 10, 6 (2018).

[10] Bander Ali Saleh Al-rimy, Mohd Aizaini Maarof, and Syed Zainudeen Mohd Shaid. 2019. Crypto-ransomware early detection model using novel incremental bagging with enhanced semi-random subspace selection. *Future Generation Computer Systems* 101 (2019), 476–491.

[11] Manaar Alam, Sarani Bhattacharya, Swastika Dutta, Sayan Sinha, Debdeep Mukhopadhyay, and Anupam Chattopadhyay. 2019. RATAFIA: Ransomware analysis using time and frequency informed autoencoders. In *2019 IEEE International Symposium on Hardware Oriented Security and Trust (HOST'19)*. IEEE Computer Society, 218–227.

[12] Ahmad O. Almashhadani, Mustafa Kaiiali, Sakir Sezer, and Philip O'Kane. 2019. A multi-classifier network-based crypto ransomware detection system: A case study of Locky ransomware. *IEEE Access* 7 (2019), 47053–47067.

[13] Samah Alsoghyer and Iman Almomani. 2019. Ransomware detection system for android applications. *Electronics* 8, 8 (2019), 868.

[14] Samah Alsoghyer and Iman Almomani. 2020. On the effectiveness of application permissions for android ransomware detection. In *2020 6th Conference on Data Science and Machine Learning Applications (CDMA'20)*. IEEE, 94–99.

[15] Abdulrahman Alzahrani, Hani Alshahrani, Ali Alshehri, and Huirong Fu. 2019. An intelligent behavior-based ransomware detection system for android platform. In *2019 1st IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA'19)*. IEEE, 28–35.

[16] Abdulrahman Alzahrani, Ali Alshehri, Hani Alshahrani, Raed Alharthi, Huirong Fu, Anyi Liu, and Ye Zhu. 2018. RanDroid: Structural similarity approach for detecting ransomware applications in android platform. In *2018 IEEE International Conference on Electro/Information Technology (EIT'18)*. IEEE, 0892–0897.

[17] Or Ami, Yuval Elovici, and Danny Hendler. 2018. Ransomware prevention using application authentication-based file access control. In *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*. 1610–1619.

[18] Nicoló Andronio, Stefano Zanero, and Federico Maggi. 2015. Heldroid: Dissecting and detecting mobile ransomware. In *International Symposium on Recent Advances in Intrusion Detection*. Springer, 382–404.

[19] Amir Atapour-Abarghouei, Stephen Bonner, and Andrew Stephen McGough. 2019. A King's ransom for encryption: Ransomware classification using augmented one-shot learning and bayesian approximation. In *2019 IEEE International Conference on Big Data (Big Data'19)*. IEEE, 1601–1606.

[20] Md. Ahsan Ayub, Andrea Continella, and Ambareen Siraj. 2020. An I/O request packet (IRP) driven effective ransomware detection scheme using artificial neural network. In *2020 IEEE 21st International Conference on Information Reuse and Integration for Data Science (IRI'20)*. IEEE Computer Society, 319–324.

[21] Seong Il Bae, Gyu Bin Lee, and Eul Gyu Im. 2019. Ransomware detection using machine learning algorithms. *Concurrency and Computation: Practice and Experience* (2019), e5422.

[22] SungHa Baek, Youngdon Jung, Aziz Mohaisen, Sungjin Lee, and DaeHun Nyang. 2018. SSD-insider: Internal defense of solid-state drive against ransomware with perfect data recovery. In *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS'18)*. IEEE, 875–884.

[23] Pranshu Bajpai and Richard Enbody. 2020. Attacking key management in ransomware. *IT Professional* 22, 2 (2020), 21–27.

[24] Pranshu Bajpai and Richard Enbody. 2020. An empirical study of key generation in cryptographic ransomware. In *2020 International Conference on Cyber Security and Protection of Digital Services (Cyber Security'20)*. IEEE, 1–8.

[25] Pranshu Bajpai, Aditya K. Sood, and Richard Enbody. 2018. A key-management-based taxonomy for ransomware. In *2018 APWG Symposium on Electronic Crime Research (eCrime'18)*. IEEE, 1–12.

[26] Abubakar Bello and Alana Maurushat. 2020. Technical and behavioural training and awareness solutions for mitigating ransomware attacks. In *Computer Science On-line Conference*. Springer, 164–176.

[27] Eduardo Berrueta, Daniel Morato, Eduardo Magaña, and Mikel Izal. 2019. A survey on detection techniques for cryptographic ransomware. *IEEE Access* 7 (2019), 144925–144944.

[28] Riccardo Bortolameotti, Thijs van Ede, Marco Caselli, Maarten H. Everts, Pieter Hartel, Rick Hofstede, Willem Jonker, and Andreas Peter. 2017. DECANTeR: DEteCtion of anomalous outbound HTTP TRaffic by passive application fingerprinting. In *Proceedings of the 33rd Annual Computer Security Applications Conference*. ACM, 373–386.

[29] Marcus Botacin, Fabricio Ceschin, Ruimin Sun, Daniela Oliveira, and André Grégio. 2021. Challenges and pitfalls in malware research. *Computers & Security* 106 (2021), 102287.

[30] Ross Brewer. 2016. Ransomware attacks: Detection, prevention and cure. *Network Security* 2016, 9 (2016), 5–9.

[31] Calvin Brierley, Jamie Pont, Budi Arief, David J. Barnes, and Julio Hernandez-Castro. 2020. PaperW8: An IoT bricking ransomware proof of concept. In *Proceedings of the 15th International Conference on Availability, Reliability and Security*. 1–10.

[32] Krzysztof Cabaj, Marcin Gregorczyk, and Wojciech Mazurczyk. 2018. Software-defined networking-based crypto ransomware detection using HTTP traffic characteristics. *Computers & Electrical Engineering* 66 (2018), 353–368.

[33] Edward Cartwright, Julio Hernandez Castro, and Anna Cartwright. 2019. To pay or not: Game theoretic models of ransomware. *Journal of Cybersecurity* 5, 1 (2019), tyz009.

[34] Jason Castiglione and Dusko Pavlovic. 2019. Dynamic distributed secure storage against ransomware. *IEEE Transactions on Computational Social Systems* (2019).

[35] Jing Chen, Chiheng Wang, Ziming Zhao, Kai Chen, Ruiying Du, and Gail-Joon Ahn. 2017. Uncovering the face of android ransomware: Characterization and real-time detection. *IEEE Transactions on Information Forensics and Security* 13, 5 (2017), 1286–1300.

[36] Qian Chen and Robert A. Bridges. 2017. Automated behavioral analysis of malware: A case study of wannacry ransomware. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA'17)*. IEEE, 454–460.

[37] Qian Chen, Sheikh Rabiul Islam, Henry Haswell, and Robert A. Bridges. 2019. Automated ransomware behavior analysis: Pattern extraction and early detection. In *International Conference on Science of Cyber Security*. Springer, 199–214.

[38] Christopher J. W. Chew and Vimal Kumar. 2019. Behaviour based ransomware detection. *Proceedings of 34th International Conference on Computers and Their Applications* 58 (2019), 127–136.

[39] Aniello Cimitile, Francesco Mercaldo, Vittoria Nardone, Antonella Santone, and Corrado Aaron Visaggio. 2018. Talos: No more ransomware victims with formal methods. *International Journal of Information Security* 17, 6 (2018), 719–738.

[40] Aviad Cohen and Nir Nissim. 2018. Trusted detection of ransomware in a private cloud using machine learning methods leveraging meta-features from volatile memory. *Expert Systems with Applications* 102 (2018), 158–178.

[41] Lena Y. Connolly and David S. Wall. 2019. The rise of crypto-ransomware in a changing cybercrime landscape: Taxonomising countermeasures. *Computers & Security* 87 (2019), 101568.

[42] Mauro Conti, Ankit Gangwal, and Sushmita Ruj. 2018. On the economic significance of ransomware campaigns: A Bitcoin transactions perspective. *Computers & Security* 79 (2018), 162–189.

[43] Andrea Continella, Alessandro Guagnelli, Giovanni Zingaro, Giulio De Pasquale, Alessandro Barenghi, Stefano Zanero, and Federico Maggi. 2016. ShieldFS: A self-healing, ransomware-aware filesystem. In *Proceedings of the 32nd Annual Conference on Computer Security Applications*. ACM, 336–347.

[44] Greg Cusack, Oliver Michel, and Eric Keller. 2018. Machine learning-based detection of ransomware using SDN. In *Proceedings of the 2018 ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization*. 1–6.

[45] Alfredo Cuzzocrea, Fabio Martinelli, and Francesco Mercaldo. 2018. A novel structural-entropy-based classification technique for supporting android ransomware detection and analysis. In *2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'18)*. IEEE, 1–7.

[46] Tooska Dargahi, Ali Dehghantanha, Pooneh Nikkhah Bahrami, Mauro Conti, Giuseppe Bianchi, and Loris Benedetto. 2019. A cyber-kill-chain based taxonomy of crypto-ransomware features. *Journal of Computer Virology and Hacking Techniques* 15, 4 (2019), 277–305.

[47] Simon R. Davies, Richard Macfarlane, and William J. Buchanan. 2020. Evaluation of live forensic techniques in ransomware attack mitigation. *Forensic Science International: Digital Investigation* 33 (2020), 300979.

[48] Hossam Faris, Maria Habib, Iman Almomani, Mohammed Eshtay, and Ibrahim Aljarah. 2020. Optimizing extreme learning machines using chains of salps for efficient android ransomware detection. *Applied Sciences* 10, 11 (2020), 3706.

[49] Adrienne Porter Felt, Elizabeth Ha, Serge Egelman, Ariel Haney, Erika Chin, and David Wagner. 2012. Android permissions: User attention, comprehension, and behavior. In *Proceedings of the 8th Symposium on Usable Privacy and Security*. 1–14.

[50] Yun Feng, Chaoge Liu, and Baoxu Liu. 2017. Poster: A new approach to detecting ransomware with deception. In *38th IEEE Symposium on Security and Privacy*.

[51] Lorenzo Fernandez Maimo, Alberto Huertas Celdran, Angel L. Perales Gomez, Felix J. Garcia Clemente, James Weimer, and Insup Lee. 2019. Intelligent and dynamic ransomware spread detection and mitigation in integrated clinical environments. *Sensors* 19, 5 (2019), 1114.

[52] Alberto Ferrante, Miroslaw Malek, Fabio Martinelli, Francesco Mercaldo, and Jelena Milosevic. 2017. Extinguishing ransomware—A hybrid approach to android ransomware detection. In *International Symposium on Foundations and Practice of Security*. Springer, 242–258.

[53] Pablo L. Gallegos-Segovia, Jack F. Bravo-Torres, Víctor M. Larios-Rosillo, Paúl E. Vintimilla-Tapia, Iván F. Yuquilima-Albarado, and Juan D. Jara-Saltos. 2017. Social engineering as an attack vector for ransomware. In *2017 CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON'17)*. IEEE, 1–6.

[54] Ziya Alper Genç, Gabriele Lenzini, and Peter Y. A. Ryan. 2018. Next generation cryptographic ransomware. In *Nordic Conference on Secure IT Systems*. Springer, 385–401.

[55] Ziya Alper Genç, Gabriele Lenzini, and Peter Y. A. Ryan. 2018. No random, no ransom: A key to stop cryptographic ransomware. In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*. Springer, 234–255.

[56] Ziya Alper Genç, Gabriele Lenzini, and Peter Y. A. Ryan. 2019. NoCry: No more secure encryption keys for cryptographic ransomware. In *International Workshop on Emerging Technologies for Authorization and Authentication*. Springer, 69–85.

[57] Amirhossein Gharib and Ali Ghorbani. 2017. DNA-droid: A real-time android ransomware detection framework. In *International Conference on Network and System Security*. Springer, 184–198.

[58] J. A. Gómez-Hernández, L. Álvarez-González, and Pedro García-Teodoro. 2018. R-Locker: Thwarting ransomware action through a honeyfile-based approach. *Computers & Security* 73 (2018), 389–398.

[59] Nikolai Hampton, Zubair Baig, and Sherali Zeadally. 2018. Ransomware behavioural analysis on windows platforms. *Journal of Information Security and Applications* 40 (2018), 44–51.

[60] Md. Mahbub Hasan and Md. Mahbubur Rahman. 2017. RansHunt: A support vector machines based ransomware analysis framework with integrated feature set. In *2017 20th International Conference of Computer and Information Technology (ICCIT'17)*. IEEE, 1–7.

[61] Matthias Held and Marcel Waldvogel. 2018. Fighting ransomware with guided undo. *NISK Journal* 11 (2018).

[62] Gregory Hill and Xavier Bellekens. 2018. CryptoKnight: Generating and modelling compiled cryptographic primitives. *Information* 9, 9 (2018), 231.

[63] Manabu Hirano and Ryotaro Kobayashi. 2019. Machine learning based ransomware detection using storage access patterns obtained from live-forensic hypervisor. In *2019 Sixth International Conference on Internet of Things: Systems, Management and Security (IOTSMS'19)*. IEEE, 1–6.

[64] Sajad Homayoun, Ali Dehghantanha, Marzieh Ahmadzadeh, Sattar Hashemi, and Raouf Khayami. 2017. Know abnormal, find evil: Frequent pattern mining for ransomware threat hunting and intelligence. *IEEE Transactions on Emerging Topics in Computing* (2017).

[65] Sajad Homayoun, Ali Dehghantanha, Marzieh Ahmadzadeh, Sattar Hashemi, Raouf Khayami, Kim-Kwang Raymond Choo, and David Ellis Newton. 2019. DRTHIS: Deep ransomware threat hunting and intelligence system at the fog layer. *Future Generation Computer Systems* 90 (2019), 94–104.

[66] Toshiki Honda, Kohei Mukaiyama, Takeharu Shirai, Tetsushi Ohki, and Masakatsu Nishigaki. 2018. Ransomware detection considering user's document editing. In *2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA'18)*. IEEE, 907–914.

[67] Danny Yuxing Huang, Damon McCoy, Maxwell Matthaios Aliapoulios, Vector Guo Li, Luca Invernizzi, Elie Bursztein, Kylie McRoberts, Jonathan Levin, Kirill Levchenko, and Alex C. Snoeren. 2018. Tracking ransomware end-to-end. In *Tracking Ransomware End-to-End*. IEEE, 0.

[68] Jian Huang, Jun Xu, Xinyu Xing, Peng Liu, and Moinuddin K. Qureshi. 2017. FlashGuard: Leveraging intrinsic flash properties to defend against encryption ransomware. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 2231–2244.

[69] Gavin Hull, Henna John, and Budi Arief. 2019. Ransomware deployment methods and analysis: Views from a predictive model and human responses. *Crime Science* 8, 1 (2019), 2.

[70] Jaime Ibarra, Usman Javed Butt, Anh Do, Hamid Jahankhani, and Arshad Jamal. 2019. Ransomware impact to SCADA systems and its scope to critical infrastructure. In *2019 IEEE 12th International Conference on Global Security, Safety and Sustainability (ICGS3'19)*. IEEE, 1–12.

[71] Yong Jin, Masahiko Tomoishi, Satoshi Matsuura, and Yoshiaki Kitaguchi. 2018. A secure container-based backup mechanism to survive destructive ransomware attacks. In *2018 International Conference on Computing, Networking and Communications (ICNC'18)*. IEEE, 1–6.

[72] Sangmoon Jung and Yoojae Won. 2018. Ransomware detection method based on context-aware entropy analysis. *Soft Computing* 22, 20 (2018), 6731–6740.

[73] Meet Kanwal and Sanjeev Thakur. 2017. An app based on static analysis for android ransomware. In *2017 International Conference on Computing, Communication and Automation (ICCCA'17)*. IEEE, 813–818.

[74] Alireza Karimi and Mohammad Hosein Moattar. 2017. Android ransomware detection using reduced opcode sequence and image similarity. In *2017 7th International Conference on Computer and Knowledge Engineering (ICCKE'17)*. IEEE, 229–234.

[75] Chee Keong Ng, Sutharshan Rajasegarar, Lei Pan, Frank Jiang, and Leo Yu Zhang. 2020. VoterChoice: A ransomware detection honeypot with multiple voting framework. *Concurrency and Computation: Practice and Experience* 32, 14 (2020), e5726.

[76] Mohamed Amine Kerrich, Adnane Addaim, and Loubna Damej. 2019. Proposed solution for HID fileless ransomware using machine learning. In *International Conference on Advanced Communication Systems and Information Security*. Springer, 180–192.

[77] Masoudeh Keshavarzi and Hamid Reza Ghaffary. 2020. I2CE3: A dedicated and separated attack chain for ransomware offenses as the most infamous cyber extortion. *Computer Science Review* 36 (2020), 100233.

[78] Eleni Ketzaki, Petros Toupas, Konstantinos M. Giannoutakis, Anastasios Drosou, and Dimitrios Tzovaras. 2020. A behaviour based ransomware detection using neural network models. In *2020 10th International Conference on Advanced Computer Information Technologies (ACIT'20)*. IEEE, 747–750.

[79] Firoz Khan, Cornelius Ncube, Lakshmana Kumar Ramasamy, Seifedine Kadry, and Yunyoung Nam. 2020. A digital DNA sequencing engine for ransomware detection using machine learning. *IEEE Access* 8 (2020), 119710–119719.

[80] Amin Kharaz, Sajjad Arshad, Collin Mulliner, William Robertson, and Engin Kirda. 2016. UNVEIL: A large-scale, automated approach to detecting ransomware. In *25th USENIX Security Symposium (USENIX Security'16)*. 757–772.

[81] Amin Kharraz and Engin Kirda. 2017. Redemption: Real-time protection against ransomware at end-hosts. In *International Symposium on Research in Attacks, Intrusions, and Defenses*. Springer, 98–119.

[82] Amin Kharraz, William Robertson, Davide Balzarotti, Leyla Bilge, and Engin Kirda. 2015. Cutting the gordian knot: A look under the hood of ransomware attacks. In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*. Springer, 3–24.

[83] Dae-Youb Kim, Geun-Yeong Choi, and Ji-Hoon Lee. 2018. White list-based ransomware real-time detection and prevention for user device protection. In *2018 IEEE International Conference on Consumer Electronics (ICCE'18)*. IEEE, 1–5.

[84] Dae-Youb Kim and Ji-hoon Lee. 2020. Blacklist vs. Whitelist-Based Ransomware Solutions. *IEEE Consumer Electronics Magazine* 9, 3 (2020), 22–28.

[85] S. H. Kok, Azween Abdullah, N. Z. Jhanjhi, and Mahadevan Supramaniam. 2019. Prevention of crypto-ransomware using a pre-encryption detection algorithm. *Computers* 8, 4 (2019), 79.

[86] S. H. Kok, A. Azween, and N. Z. Jhanjhi. 2020. Evaluation metric for crypto-ransomware detection using machine learning. *Journal of Information Security and Applications* 55 (2020), 102646.

[87] Eugene Kolodenker, William Koch, Gianluca Stringhini, and Manuel Egele. 2017. PayBreak: Defense against cryptographic ransomware. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*. ACM, 599–611.

[88] Simon Kramer and Julian C. Bradfield. 2010. A general definition of malware. *Journal in Computer Virology* 6, 2 (2010), 105–114.

[89] Nir Kshetri and Jeffrey Voas. 2017. Do crypto-currencies fuel ransomware? *IT professional* 19, 5 (2017), 11–15.

[90] Jeong Kyu Lee, Seo Yeon Moon, and Jong Hyuk Park. 2017. CloudRPS: A cloud analysis based enhanced ransomware prevention system. *The Journal of Supercomputing* 73, 7 (2017), 3065–3084.

[91] Kyungroul Lee, Sun-Young Lee, and Kangbin Yim. 2019. Machine learning based file entropy analysis for ransomware detection in backup systems. *IEEE Access* 7 (2019), 110205–110215.

[92] Kyungroul Lee, Kangbin Yim, and Jung Taek Seo. 2018. Ransomware prevention technique using key backup. *Concurrency and Computation: Practice and Experience* 30, 3 (2018), e4337.

[93] Suhyeon Lee, Huy Kang Kim, and Kyounggon Kim. 2019. Ransomware protection using the moving target defense perspective. *Computers & Electrical Engineering* 78 (2019), 288–299.

[94] Tianliang Lu, Yanhui Du, Jing Wu, and Yuxuan Bao. 2019. Ransomware Detection Based on an Improved Double-Layer Negative Selection Algorithm. In *International Conference on Testbeds and Research Infrastructures*. Springer, 46–61.

[95] Tianliang Lu, Lu Zhang, Shunye Wang, and Qi Gong. 2017. Ransomware detection based on V-detector negative selection algorithm. In *2017 International Conference on Security, Pattern Analysis, and Cybernetics (SPAC'17)*. IEEE, 531–536.

[96] Ewa Luger, Stuart Moran, and Tom Rodden. 2013. Consent for all: Revealing the hidden complexity of terms and conditions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2687–2696.

[97] Xin Luo and Qinyu Liao. 2007. Awareness education as the key to ransomware prevention. *Information Systems Security* 16, 4 (2007), 195–202.

[98] Davide Maiorca, Francesco Mercaldo, Giorgio Giacinto, Corrado Aaron Visaggio, and Fabio Martinelli. 2017. R-PackDroid: API package-based characterization and detection of mobile ransomware. In *Proceedings of the Symposium on Applied Computing*. 1718–1723.

[99] Sumith Maniath, Aravind Ashok, Prabaharan Poornachandran, V. G. Sujadevi, A. U. Prem Sankar, and Srinath Jan. 2017. Deep learning LSTM based ransomware detection. In *2017 Recent Developments in Control, Automation & Power Engineering (RDCAPE'17)*. IEEE, 442–446.

[100] Steve Mansfield-Devine. 2017. Fileless attacks: Compromising targets without malware. *Network Security* 2017, 4 (2017), 7–11.

[101] Michael J. May and Etamar Laron. 2019. Combating ransomware using content analysis and complex file events. In *2019 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS'19)*. IEEE, 1–5.

[102] Faustin Mbol, Jean-Marc Robert, and Alireza Sadighian. 2016. An efficient approach to detect torrentlocker ransomware in computer systems. In *International Conference on Cryptology and Network Security*. Springer, 532–541.

[103] Timothy McIntosh, Julian Jang-Jaccard, Paul Watters, and Teo Susnjak. 2019. The inadequacy of entropy-based ransomware detection. In *International Conference on Neural Information Processing*. Springer, 181–189.

[104] Timothy McIntosh, Julian Jang-Jaccard, Paul Watters, and Teo Susnjak. 2019. Masquerade attacks against security software exclusion lists. (2019), 5–12.

[105] Timothy McIntosh, Paul Watters, A. S. M. Kayes, Alex Ng, and Yi-Ping Phoebe Chen. 2020. Enforcing situation-aware access control to build malware-resilient file systems. *Future Generation Computer Systems* 115 (2020), 568–582.

[106] Timothy R. McIntosh, Julian Jang-Jaccard, and Paul A. Watters. 2018. Large scale behavioral analysis of ransomware attacks. In *International Conference on Neural Information Processing*. Springer, 217–229.

[107] May Medhat, Samir Gaber, and Nashwa Abdelbaki. 2018. A new static-based framework for ransomware detection. In *2018 IEEE 16th International Conference on Dependable, Autonomic and Secure Computing, 16th International Conference on Pervasive Intelligence and Computing, 4th International Conference on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech'18)*. IEEE, 710–715.

[108] Shagufta Mehnaz, Anand Mudgerikar, and Elisa Bertino. 2018. RWGuard: A real-time detection system against cryptographic ransomware. In *International Symposium on Research in Attacks, Intrusions, and Defenses*. Springer, 114–136.

[109] Per Håkon Meland, Yara Fareed Fahmy Bayoumy, and Guttorm Sindre. 2020. The Ransomware-as-a-Service economy within the darknet. *Computers & Security* (2020), 101762.

[110] Francesco Mercaldo, Vittoria Nardone, Antonella Santone, and Corrado Aaron Visaggio. 2016. Ransomware steals your phone. formal methods rescue it. In *International Conference on Formal Techniques for Distributed Objects, Components, and Systems*. Springer, 212–221.

[111] Donghyun Min, Donggyu Park, Jinwoo Ahn, Ryan Walker, Junghee Lee, Sungyong Park, and Youngjae Kim. 2018. Amoeba: An autonomous backup and recovery SSD for ransomware attack defense. *IEEE Computer Architecture Letters* 17, 2 (2018), 245–248.

[112] Jaimin Modi, Issa Traore, Asem Ghaleb, Karim Ganame, and Sherif Ahmed. 2019. Detecting ransomware in encrypted web traffic. In *International Symposium on Foundations and Practice of Security*. Springer, 345–353.

[113] Chris Moore. 2016. Detecting ransomware with honeypot techniques. In *2016 Cybersecurity and Cyberforensics Conference (CCC'16)*. IEEE, 77–81.

[114] Daniel Morato, Eduardo Berrueta, Eduardo Magaña, and Mikel Izal. 2018. Ransomware early detection by the analysis of file sharing traffic. *Journal of Network and Computer Applications* 124 (2018), 14–32.

[115] Andreas Moser, Christopher Kruegel, and Engin Kirda. 2007. Limits of static analysis for malware detection. In *23rd Annual Computer Security Applications Conference (ACSAC'07)*. IEEE, 421–430.

[116] Ori Or-Meir, Nir Nissim, Yuval Elovici, and Lior Rokach. 2019. Dynamic malware analysis in the modern era—A state of the art survey. *ACM Computing Surveys (CSUR)* 52, 5 (2019), 1–48.

[117] Joon-young Paik, Joong-Hyun Choi, Rize Jin, Jianming Wang, and Eun-Sun Cho. 2019. Buffer management for identifying crypto-ransomware attack in environment with no semantic information. In *2019 IEEE International Conference on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom'19)*. IEEE, 443–450.

[118] Aurélien Palisse, Antoine Durand, Hélène Le Bouder, Colas Le Guernic, and Jean-Louis Lanet. 2017. Data aware defense (DaD): Towards a generic and practical ransomware countermeasure. In *Nordic Conference on Secure IT Systems*. Springer, 192–208.

[119] Aurélien Palisse, Hélène Le Bouder, Jean-Louis Lanet, Colas Le Guernic, and Axel Legay. 2016. Ransomware and the legacy crypto API. In *International Conference on Risks and Security of Internet and Systems*. Springer, 11–28.

[120] Jamie Pont, Budi Arief, and Julio Hernandez-Castro. 2020. Why current statistical approaches to ransomware detection fail. In *International Conference on Information Security*. Springer, 199–216.

[121] Jamie Pont, Osama Abu Oun, Calvin Brierley, Budi Arief, and Julio Hernandez-Castro. 2019. A roadmap for improving the impact of anti-ransomware research. In *Nordic Conference on Secure IT Systems*. Springer, 137–154.

[122] Mila Dalla Preda, Mihai Christodorescu, Somesh Jha, and Saumya Debray. 2007. A semantics-based approach to malware detection. *ACM SIGPLAN Notices* 42, 1 (2007), 377–388.

[123] James H. Price and Judy Murnan. 2004. Research limitations and the necessity of reporting them. *American Journal of Health Education* 35, 2 (2004), 66.

[124] Florian Quinkert, Thorsten Holz, K. S. M. Hossain, Emilio Ferrara, and Kristina Lerman. 2018. RAPTOR: Ransomware attack PredicTOR. arXiv:1803.01598.

[125] Gowtham Ramesh and Anjali Menen. 2020. Automated dynamic approach for detecting ransomware using finite-state machine. *Decision Support Systems* 138 (2020), 113400.

[126] Rahul Rastogi, Gaurav Agarwal, and R. K. Shukla. Interactive security of ransomware with heuristic random bit generator. In *ICCCE 2020*. Springer, 965–973.

[127] Bheemidi Vikram Reddy, Gutha Jaya Krishna, Vadlamani Ravi, and Dipankar Dasgupta. 2020. Machine learning and feature selection based ransomware detection using hexacodes. In *Evolution in Computational Intelligence*. Springer, 583–597.

[128] Talia Ringer, Dan Grossman, and Franziska Roesner. 2016. Audacious: User-driven access control with unmodified operating systems. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. 204–216.

[129] Krishna Chandra Roy and Qian Chen. 2020. DeepRan: Attention-based BiLSTM and CRF for ransomware early detection and classifcation. *Information Systems Frontiers* (2020), 1–17.

[130] Nolen Scaife, Henry Carter, Patrick Traynor, and Kevin R. B. Butler. 2016. Cryptolock (and drop it): Stopping ransomware attacks on user data. In *2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS'16)*. IEEE, 303–312.

[131] Michele Scalas, Davide Maiorca, Francesco Mercaldo, Corrado Aaron Visaggio, Fabio Martinelli, and Giorgio Giacinto. 2019. On the effectiveness of system API-related information for Android ransomware detection. *Computers & Security* 86 (2019), 168–182.

[132] Daniele Sgandurra, Luis Muñoz-González, Rabih Mohsen, and Emil C. Lupu. 2016. Automated dynamic analysis of ransomware: Benefits, limitations and use for detection. arXiv:1609.03020.

[133] Shaila Sharmeen, Yahye Abukar Ahmed, Shamsul Huda, Bari Ş. Koçer, and Mohammad Mehedi Hassan. 2020. Avoiding future digital extortion through robust protection against ransomware threats using deep learning based adaptive approaches. *IEEE Access* 8 (2020), 24522–24534.

[134] Saiyed Kashif Shaukat and Vinay J. Ribeiro. 2018. RansomWall: A layered defense system against cryptographic ransomware attacks using machine learning. In *2018 10th International Conference on Communication Systems & Networks (COMSNETS'18)*. IEEE, 356–363.

[135] Jeffrey Shirley and David Evans. 2008. The user is not the enemy: Fighting malware by tracking user intentions. In *Proceedings of the 2008 New Security Paradigms Workshop*. 33–45.

[136] Ali Shuja Siddiqui, Chia-Che Lee, and Fareena Saqib. 2017. Hardware based protection against malwares by PUF based access control mechanism. In *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS'17)*. IEEE, 1312–1315.

[137] Sanggeun Song, Bongjoon Kim, and Sangjun Lee. 2016. The effective ransomware prevention technique using process monitoring on android platform. *Mobile Information Systems* 2016 (2016).

[138] Dan Su, Jiqiang Liu, Xiaoyang Wang, and Wei Wang. 2018. Detecting android locker-ransomware on Chinese social networks. *IEEE Access* 7 (2018), 20381–20393.

[139] Kul Prasad Subedi, Daya Ram Budhathoki, Bo Chen, and Dipankar Dasgupta. 2017. RDS3: Ransomware defense strategy by using stealthily spare space. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI'17)*. IEEE, 1–8.

[140] Kul Prasad Subedi, Daya Ram Budhathoki, and Dipankar Dasgupta. 2018. Forensic analysis of ransomware families using static and dynamic analysis. In *2018 IEEE Security and Privacy Workshops (SPW'18)*. IEEE, 180–185.

[141] Yuki Takeuchi, Kazuya Sakai, and Satoshi Fukumoto. 2018. Detecting ransomware using support vector machines. In *Proceedings of the 47th International Conference on Parallel Processing Companion*. 1–6.

[142] Fei Tang, Boyang Ma, Jinku Li, Fengwei Zhang, Jipeng Su, and Jianfeng Ma. 2020. RansomSpector: An introspection-based approach to detect crypto ransomware. *Computers & Security* (2020), 101997.

[143] Jason Thomas. 2018. Individual cyber security: Empowering employees to resist spear phishing to prevent identity theft and ransomware attacks. *International Journal of Business Management* 12, 3 (2018), 1–23.

[144] Aragorn Tseng, Y. Chen, Y. Kao, and T. Lin. 2016. Deep learning for ransomware detection. *IEICE Technical Report* 116, 282 (2016), 87–92.

[145] Hasan Turaev, Pavol Zavarsky, and Bobby Swar. 2018. Prevention of ransomware execution in enterprise environment on windows OS: Assessment of application whitelisting solutions. In *2018 1st International Conference on Data Intelligence and Security (ICDIS'18)*. IEEE, 110–118.

[146] Mayank Verma, Ponnurangam Kumarguru, Shuva Brata Deb, and Anuradha Gupta. 2018. Analysing indicator of compromises for ransomware: Leveraging IOCs with machine learning techniques. In *2018 IEEE International Conference on Intelligence and Security Informatics (ISI'18)*. IEEE, 154–159.

[147] Peiying Wang, Shijie Jia, Bo Chen, Luning Xia, and Peng Liu. 2019. MimosaFTL: Adding secure and practical ransomware defense strategy to flash translation layer. In *Proceedings of the 9th ACM Conference on Data and Application Security and Privacy*. 327–338.

[148] ZiHan Wang, ChaoGe Liu, Jing Qiu, ZhiHong Tian, Xiang Cui, and Shen Su. 2018. Automatically traceback RDP-based targeted ransomware attacks. *Wireless Communications and Mobile Computing* 2018 (2018).

[149] Azka Wani and S. Revathi. 2020. Ransomware protection in loT using software defined networking. *International Journal of Electrical & Computer Engineering (2088-8708)* 10 (2020).

[150] Mattias Weckstén, Jan Frick, Andreas Sjöström, and Eric Järpe. 2016. A novel method for recovery from Crypto Ransomware infections. In *2016 2nd IEEE International Conference on Computer and Communications (ICCC'16)*. IEEE, 1354–1358.

[151] Tianda Yang, Yu Yang, Kai Qian, Dan Chia-Tien Lo, Ying Qian, and Lixin Tao. 2015. Automated detection and analysis for android ransomware. In *2015 IEEE 17th International Conference on High Performance Computing and Communications, 2015 IEEE 7th International Symposium on Cyberspace Safety and Security, and 2015 IEEE 12th International Conference on Embedded Software and Systems*. IEEE, 1338–1343.

[152] Adam Young and Moti Yung. 1996. Cryptovirology: Extortion-based security threats and countermeasures. In *Proceedings 1996 IEEE Symposium on Security and Privacy*. IEEE, 129–140.

[153] Bin Zhang, Wentao Xiao, Xi Xiao, Arun Kumar Sangaiah, Weizhe Zhang, and Jiajia Zhang. 2020. Ransomware classification using patch-based CNN and self-attention network on embedded N-grams of opcodes. *Future Generation Computer Systems* 110 (2020), 708–720.

[154] Bin Zhang, Wentao Xiao, Xi Xiao, Arun Kumar Sangaiah, Weizhe Zhang, and Jiajia Zhang. 2020. Ransomware classification using patch-based CNN and self-attention network on embedded N-grams of opcodes. *Future Generation Computer Systems* 110 (2020), 708–720.

[155] Hanqi Zhang, Xi Xiao, Francesco Mercaldo, Shiguang Ni, Fabio Martinelli, and Arun Kumar Sangaiah. 2019. Classification of ransomware families with machine learning based on N-gram of opcodes. *Future Generation Computer Systems* 90 (2019), 211–221.

[156] Aaron Zimba, Zhaoshun Wang, and Hongsong Chen. 2018. Multi-stage crypto ransomware attacks: A new emerging cyber threat to critical infrastructure and industrial control systems. *ICT Express* 4, 1 (2018), 14–18.

[157] Hiba Zuhair and Ali Selamat. 2019. RANDS: A machine learning-based anti-ransomware tool for windows platforms. *Frontiers in Artificial Intelligence and Applications* 318 (2019).

[158] Hiba Zuhair, Ali Selamat, and Ondrej Krejcar. 2020. A multi-tier streaming analytics model of 0-day ransomware detection using machine learning. *Applied Sciences* 10, 9 (2020), 3210.

Received February 24, 2021; revised June 16, 2021; accepted July 26, 2021